MASTER EN SCIENCES INFORMATIQUES

Analyse d'un modèle mathématique d'un protocole de diffusion de rumeur

WELCOMME, Olivier

Award date:
2021

Awarding institution:
Universite de Namur

UNIVERSITE DE NAMUR

FACULTÉ D'INFORMATIQUE

Analyse d'un modèle mathématique d'un protocole de diffusion de rumeur

Mémoire présenté pour l'obtention
du grade académique en master informatique en horaire décalé

Olivier WELCOMME

Juin 2021

Analyse d'un modèle mathématique d'un protocole de diffusion de rumeur

Olivier WELCOMME

Promoteur : Pr. Marie-Ange Remiche

Abstract

L'objectif de ce mémoire est d'introduire le protocole de diffusion de rumeur. Nous souhaitons effectuer une analyse du protocole à travers une modélisation mathématique. Cette analyse a pour but d'être suffisante à la création d'un modèle de simulation pouvant servir à une éventuelle implémentation du protocole. Deux versions du protocoles sont abordées : les protocoles de diffusion de rumeur à temps discret et à temps continu. Nous souhaitons également proposer une modification du protocole de diffusion de rumeur à temps continu, en lui ajoutant un processus appelé Markovian Arrival Process (MAP). Une introduction à ce processus est donc effectuée, et est accompagnée d'un modèle de simulation proposant une manière de simuler ce Markovian Arrival Process.

This work will be centered around an introduction to the rumor spreading protocol. The aim of this work is to provide a mathematical analysis of the protocol. This analysis will be performed on two versions of the protocol : the continuous-time rumor spreading protocol, and the discrete-time rumor spreading protocol. The aim of this analysis is to provide a mathematical model allowing us to create a computational model of a simulation of the protocol. This computational model is followed by the results acquired through an implementation of the simulation. Another goal of this work is to offer a modification to the continuous-time rumor spreading protocol, by incorporing a process called Markovian Arrival Process (MAP). An introduction of this process is done, followed by a computational model of the simulation of this Markovian Arrival Process.

Key words :

Rumor Spreading, Population Protocol, Markovian Arrival Process, Gossip Spreading, MAP

Table des matières

Chapitre 1

Introduction

L'objectif de ce mémoire est de proposer une approche du protocole de diffusion de rumeur. Cette approche est effectuée par une modélisation mathématique du protocole, nous permettant ensuite de réaliser un modèle de simulation du protocole. Deux variantes du protocole de diffusion de rumeur sont abordées. La première est le protocole de diffusion de rumeur en temps discret, et la seconde est le protocole en temps continu. Le modèle de simulation proposé dans ce mémoire est accompagné des résultats obtenus lors de l'implémentation d'une simulation du protocole.

Un second objectif de ce travail est l'utilisation du Markovian Arrival Process (MAP) dans le cadre du protocole de diffusion de rumeur en temps continu. Dans ce cadre, nous souhaitons décrire le MAP. Cette description est réalisée avec une introduction du processus de renouvellement, de la distribution de type phase, et du processus de renouvellement de type phase. Nous proposons ensuite un modèle permettant la simulation d'un Markovian Arrival Process, et introduisons son usage dans le cas des protocoles de diffusion de rumeur.

Le mémoire est divisé en plusieurs chapitres.

Dans un premier chapitre, nous mettons en évidence certaines situations justifiant l'usage d'un protocole de diffusion de rumeur.

Dans un second chapitre, nous introduisons les protocoles de population, et effectuons une classification de ces derniers. Cette classification nous permet de souligner le protocole de diffusion de rumeur qui fait l'objet de ce travail.

Dans le troisième chapitre, nous analysons le protocole de diffusion de rumeur à l'aide d'un modèle mathématique. Cette analyse est effectuée pour les protocoles en temps discret et en temps continu. Pour l'analyse du protocole en temps discret, nous développons également le calcul de la distribution du temps de convergence.

Dans le quatrième chapitre, nous proposons un modèle permettant l'implémentation d'une simulation du protocole de diffusion de rumeur en temps discret. Dans ce chapitre, nous analysons également les résultats issus d'une telle simulation, nous permettant d'appuyer l'analyse mathématique effectuée dans le chapitre précédent.

Dans le cinquième chapitre, nous souhaitons aborder le Markovian Arrival Process. Cela est effectué par une introduction des processus de renouvellement ainsi que de la distribution de type phase.

Dans le sixième chapitre est proposé un modèle qui nous permet de simuler le Markovian Arrival Process.

Chapitre 2

Motivation historique

Dans ce chapitre, il est question de mettre en évidence quelques situations qui prouvent l'intérêt du développement des protocoles de diffusion de rumeur, par des applications en informatique.

2.1 Historique

2.1.1 Propagation de mise à jour de base de données

Initialement, le protocole de diffusion de rumeur dans un cadre informatique apparait en 1989, introduit par Alan Demers et al [4]. A l'origine, les auteurs étaient confrontés au problème suivant : on a une base de données que l'on souhaite exporter sur un grand nombre de sites différents. Dès lors, lorsqu'une mise à jour de la base de données initiale est effectuée, on peut voir apparaitre un problème vis-à-vis de la répercussion de cette mise à jour sur toutes ses répliques. Le papier de Demers et al. vise alors à proposer divers algorithmes aléatoires[1] très simples afin de propager les mises à jour à travers les répliques de la base de données. Chacun de ces algorithmes présente alors peu de contraintes vis-à-vis du réseau de communication sur lequel se développe le système. L'objectif de cet algorithme est d'atteindre un état où toutes les mises à jour sont répercutées sur toutes les répliques. Dans le cadre de ce papier, une première analogie est effectuée avec les phénomènes étudiés en épidémiologie, car chacun des algorithmes a un comportement très proche de celui d'une épidémie.

1. Un algorithme est dit probabiliste si son comportement dépend à la fois des données du problème et de valeurs produites par un générateur de nombres aléatoires.

L'étude de Demers et al. est portée principalement sur trois méthodes différentes que nous expliquons.

La première méthode proposée par Demers et al. est le système par mails directs (Direct mail), où chaque nouvel update est immédiatement communiqué depuis le site d'origine sur tous les autres sites. Bien que ce modèle soit très efficace en terme de temps de propagation, il n'est en réalité pas entièrement fiable, car certains sites individuels ne connaissent pas toujours tous les autres sites, et certains mails sont parfois perdus.

Ensuite, nous avons la méthode "Anti-entropie", où chaque site choisit régulièrement un autre site de manière aléatoire, et échange le contenu de sa base de données avec l'autre site, en cherchant ainsi les conflits trouvés entre les deux bases de données. Bien qu'ici la fiabilité de la méthode soit très bonne, il demande d'examiner le contenu des bases de données, et ne peut dès lors pas être appliqué trop fréquemment. De plus, cette méthode montre un temps de propagation bien plus lent que celui de la méthode Direct mail, même si sa fiabilité est très grande.

Finalement, la dernière méthode est celle de "diffusion de rumeur", nous intéressant ici dans le cadre de ce mémoire. Dans la diffusion de rumeur, on considère que tous les sites sont initialement ignorants. Quand un site reçoit un update de sa base de données, cet update est interprété comme une rumeur. Cet update est périodiquement communiqué à un autre site choisi de manière aléatoire et en s'assurant que les changements aient été effectués sur l'autre site. De plus, dans le modèle de Demers et al, on envisage qu'un site ne considère plus la mise à jour comme une rumeur après un certain temps, et retient alors son état sans le propager davantage. Cette méthode est généralement plus utilisée que la méthode anti-entropie. Cependant, dans un système où on considère que la rumeur n'est plus propagée après un certain temps, il demeure une probabilité que la mise à jour n'atteigne pas tous les sites.

Un parallèle peut être effectué avec les processus épidémiques, pour la diffusion de rumeur et la méthode anti-entropie. En effet, les résultats mathématiques obtenus dans la théorie épidémiologique peuvent y être appliqués.

L'étude du problème de la mise à jour de la base de données a surtout été motivée par une étude qui avait été effectuée à l'époque sur les serveurs "Clearinghouse" de la CIN (Xerox Corporate Internet). Le réseau CIN comprenait des centaines de connexions Ethernet connectées par des routeurs ainsi que des lignes téléphoniques. Des milliers de serveurs et d'ordinateurs étaient connectés à ce réseau, et un paquet venant du

Japon jusqu'en Europe devait par exemple traverser environ 14 routeurs et 7 lignes téléphoniques. Les serveurs étudiés (serveurs Clearinghouse) étaient en charge de gérer les transitions à travers les différentes couches du réseau. Pour cela, les domaines étaient sauvegardés dans un ou plusieurs serveurs Clearinghouse, ces serveurs se comptant par centaines.

Dès lors, les trois approches citées plus haut ont été abordées dans l'optique de trouver une solution à ce problème. Au final, la solution qui a été implémentée est celle de l'anti-entropie, qui a présenté une hausse impressionnante des performances, et ainsi la réduction du trafic sur le réseau. Cependant, il est mentionné qu'il pourrait être plus adapté de combiner la diffusion de rumeur avec le modèle anti-entropie. L'idée serait d'avoir le modèle de diffusion de rumeur classique, en se servant du modèle anti-entropie comme back-up afin d'atteindre les serveurs qui n'ont pas été atteints par la diffusion de rumeur.

2.1.2 Découverte de réseaux

Outre la mise à jour de bases de données, ce type de protocole peut également se montrer utile par exemple dans des systèmes de découverte de réseaux. Un système de découverte de réseau peut être résumé ainsi : dans de larges réseaux, des ordinateurs doivent prendre conscience les uns des autres afin de communiquer. Le principe de la découverte de réseau repose sur la manière par laquelle les ordinateurs prennent conscience de l'existence des autres ordinateurs.

En 1999, M. Harchol-Baiter, T. Leightont et D. Lewid publient un papier [5] axé autour du la situation suivante : dans de très larges réseaux d'ordinateurs, il n'est pas rare que deux ordinateurs soient amenés à coopérer pour la réalisation d'une tâche. Pour que cela soit possible, il faut dès lors que chacun des deux ordinateurs aient conscience de l'existence de l'autre. L'objectif des auteurs du papier est alors d'examiner des algorithmes afin que les ordinateurs d'un réseau prennent conscience des autres ordinateurs du réseau. Cette prise de conscience est permise à l'aide de requêtes effectuées aux ordinateurs dont ils connaissent déjà l'existence. Ces algorithmes sont donc basés uniquement sur l'échange de requêtes entre les ordinateurs qui se connaissent.

Sur les différents algorithmes étudiés, celui retenu est celui appelé "Name-Dropper". L'idée est assez simple, et est exactement la même que celle introduite par Deemers et al. : de manière régulière, un ordinateur transmet la liste des ordinateurs qu'il connait à un ordinateur voisin choisi aléatoirement. Les ordinateurs connus par ce

voisin deviennent alors ceux transmis combinés à ceux qu'il connait déjà. Le nom de cet algorithme (Name-Dropper) provient du phénomène social "name dropping" où un nouveau venu s'incruste dans un groupe de personnes, mentionnant le nom des personnes qu'il connait en espérant pouvoir en tirer profit.

2.1.3 Agrégation de données

D'autres études, comme l'étude de D. Kempe, A. Dobra, et J. Gehrke effectuée en 2003 [8], ont prouvé l'utilité que peut avoir ce genre de méthodes dans des architectures distribuées à grande échelle. Effectivement, lors de la réalisation de ce papier, les dernières années ont été marquées par une grande révolution vis-à-vis du développement de la connectivité entre les ordinateurs, faisant évoluer les systèmes d'un modèle centralisé à un système grandement distribué. Cette hausse d'échelle apporte avec elle des instabilités, et les problèmes apparaissant dans les nœuds deviennent alors de plus en plus présents. Dès lors, l'apparition de protocoles décentralisés basés sur la propagation de rumeur a commencé à se faire remarquer. Cela était dans le but d'avoir une approche qui maintient une certaine simplicité et une adaptation à l'échelle du système, en maintenant une bonne fiabilité. Cette étude de 2003 vient pointer du doigt le coeur du problème : la vitesse de diffusion de l'information, en analysant cette vitesse sur des réseaux présentant des faiblesses sur des nœuds, ou sur des mécanismes basés sur "l'inondation" de l'information dans le système.

De plus, ce papier vient étudier le principe d'agrégation de nœuds dans un tel réseau distribué. Le problème d'agrégation de nœuds consiste à calculer une fonction "d'agrégation" (sommes, moyennes, etc...), appliquée sur les valeurs associées aux différents nœuds. Ainsi, Kempe, Dobra et Gehrke cherchent à utiliser une méthode décentralisée et fiable afin de pouvoir calculer la valeur de ces fonctions, en utilisant uniquement des petits messages. Ils utilisent alors un protocole basé sur la diffusion de rumeur afin d'arriver à cet objectif.

Ce problème d'estimation de fonction d'agrégation avait été initialement introduit également en 2003 par M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani [9]. L'émergence des réseaux pair à pair (P2P) amenait avec elle le besoin de collecter un tas de valeurs statistiques vis-à-vis de la participation des différents nœuds. Or, ces réseaux ne présentaient pas beaucoup d'outils afin de pouvoir collecter les valeurs de ces fonctions sur les données agrégées. Le papier de Kempe, Dobra et Gehrke qui vient d'être mentionné a donc pour objectif d'utiliser les mécanismes de diffusion de rumeur pour améliorer ce qui est suggéré dans l'étude de Bawa et al. introduisant le problème.

2.1.4 Autres applications : la diffusion d'un virus

Le modèle employé dans le cadre de la diffusion de rumeur est en réalité employé en dehors du domaine informatique depuis bientôt plus d'un siècle. Effectivement, il a été par exemple utilisé en 1927 par Kermack et al. [7] afin de modéliser un phénomène réel : la diffusion de maladies infectieuses. Cela a été également historiquement la première science à s'intéresser au phénomène utilisé dans la diffusion de rumeur.

L'objectif de cet article était centré autour d'une des plus grandes difficultés en épidémiologie de l'époque : la difficulté de trouver un facteur causal paraissant être suffisant que pour présenter une magnitude dans l'épidémie telle que la quasi totalité de la population serait touchée par la maladie. Ainsi, le problème est résumé de la manière suivante : une personne ou plus est infectée et introduite dans une communauté d'individus, plus ou moins susceptibles d'attraper la maladie. La maladie se propage, allant des personnes infectées aux personnes non-infectées lors d'un contact, et chaque personne infectée finit par soit mourir, soit être rétabli.

Dans le cadre de l'étude, il est choisi de considérer que chaque individu a une chance égale d'attraper la maladie, et qu'une infection implique une immunité totale aux futures infections de la même maladie. Le temps est divisé en plusieurs intervalles réguliers, l'infection prenant instantanément place lors de la transition entre chaque intervalle de temps.

Plus récemment, la diffusion de rumeur a été prise en considération afin de modéliser la propagation de virus dans un réseau informatique. En 2005, N. Berger, C. Borgs, J. T. Chayes, et A. Saberi ont effectué une étude [1] afin d'analyser la propagation d'un virus sur Internet à partir d'un processus de "contacts" sur un graphe généré aléatoirement. Ils montrent dans leur étude que n'importe quel virus associé à un taux d'infection positif a une chance de devenir épidémique.

Étant donné les preuves que les réseaux comme Internet présentent une architecture qui évolue de manière exponentielle, il est important de réaliser une étude de leurs propriétés structurelles. L'étude consiste dès lors à trouver une modélisation de la propagation d'infections virales sur ce type de réseaux.

Le principe employé pour cette étude est à nouveau un principe similaire à ceux vus précédemment : les différents nœuds choisissent un voisin aléatoirement, de manière uniforme, afin de symboliser la transmission du virus. Ils mentionnent également la possibilité de combiner la probabilité uniforme avec un système d'attachement préfé-

rentiel, cela étant utile pour plusieurs analyses structurelles et des analyses des propriétés dynamiques des graphes générés par attachement préférentiel.

2.2 Conclusion

Ainsi, il existe de nombreuses études qui ont montré un intérêt pour la diffusion de rumeur. La majeure question qui motive généralement l'utilisation du protocole de diffusion de rumeur est la suivante : quel est le temps nécessaire pour que tous les nœuds du système soient informés de la rumeur. On appelle ce temps *le temps de convergence*, qui par essence est aléatoire.

Yves Mocquard étudie dans sa thèse [11] l'impact du choix aléatoire et de l'instant de diffusion sur la distribution du temps de convergence. Il propose des résultats analytiques pour expliciter ces distributions. Dans la suite de ce mémoire, nous présentons de manière plus formelle une classification des protocoles de population. Cette classification a pour objectif d'introduire clairement le protocole sur lequel nous travaillons ensuite dans le chapitre 4, à savoir le protocole de diffusion de rumeur.

Chapitre 3

Classification des protocoles de population

Dans ce chapitre, nous présentons une classification de différents protocoles de population. Cette classification ne se veut pas exhaustive, et a pour but d'introduire clairement le protocole de population sur lequel nous travaillons ensuite dans le chapitre suivant.

Dans la première section, nous introduisons de manière formelle ce qu'est un modèle de protocole de population, afin de pouvoir débuter leur classification.

Dans la seconde section, nous classifions différents protocoles de population en les différenciant à l'aide de certaines caractéristiques qui leur sont associées.

La troisième section a pour but de mettre en évidence des contraintes d'implémentation pouvant apparaître lors de la mise en place d'un protocole de population.

Finalement, la dernière section précise le protocole sur lequel le reste du mémoire est centré, à savoir le protocole de diffusion de rumeur.

3.1 Protocole de population

Cette section introduit ce qu'est un protocole de population, son formalisme, et son utilité. Cette section se base principalement sur la description des protocoles de population effectuée par Yves Mocquard [6].

Angluin et al. proposent en 2008 un modèle formel du protocole de population [3].Un modèle de protocole de population est un modèle abstrait décrivant un réseau de capteurs et la manière dont ceux-ci sont liés entre eux. Dans ce modèle, les capteurs interagissent par paire. Un capteur est considéré comme une entité qui peut connaître une information. Un capteur est susceptible de communiquer l'information qu'il connait à d'autres capteurs, ou à une entité extérieure souhaitant connaitre son information en l'interrogeant. Cet ensemble de capteurs liés entre eux est ce qu'on appelle un réseau. Le modèle de protocole de population est donc une abstraction de ce réseau et des capteurs qu'il contient. Le réseau est représenté par ce qu'on appelle un graphe, où les capteurs sont les nœuds de ce graphe.

Ainsi, les nœuds du graphe sont chacun associés à une valeur, correspondant à l'information connue par le capteur. Lors des interactions entre les nœuds, cette information est communiquée entre les nœuds. A la suite d'une interaction, un nœud est susceptible de voir l'information qu'il connait mise à jour.

Si nous avons un ensemble de n nœuds, nous disons que la taille du graphe est n. Les nœuds interagissent par paire de manière séquentielle. Ainsi, de manière séquentielle, un ou plusieurs nœuds sont choisis de manière aléatoire afin de procéder à une interaction avec un autre nœud voisin lui-même choisi de manière aléatoire. Au cours de cette interaction, l'information connue par les nœuds qui interagissent est communiquée, et potentiellement mise à jour. Ainsi, les valeurs associées aux nœuds évoluent à mesure que les interactions s'enchaînent, de manière séquentielle.

Pour le bon fonctionnement du protocole de population, il est nécessaire que l'état de chaque nœud (la valeur associée) soit initialement défini. Cet état initial est appelé état d'entrée. A tout moment, nous sommes susceptibles d'interroger un des nœuds du graphe. La réponse du nœud interrogé est dépendante de son état, et est appelée état de sortie. En général, l'état de sortie correspond à la valeur du nœud au moment où celui-ci est interrogé.

Au fil des interactions, il est possible qu'un graphe atteigne un stade tel que le graphe n'évolue plus. On dit alors que le graphe a atteint un état de convergence. Par dé-

finition, l'état de convergence est l'état du graphe où toutes les réponses des nœuds (c'est-à-dire tous les états de sortie) correspondent à l'information attendue. L'information attendue peut varier en fonction du type de protocole de population choisi, et est abordée lors de la classification.

On appelle temps de convergence le temps qu'il faut à partir de l'état initial du graphe pour que le graphe atteigne l'état de convergence. Étant donné que les interactions se font de manière séquentielle, le temps peut être exprimé de multiples manières. Cette notion de temps peut être définie par exemple par le nombre d'interactions effectuées entre tous les nœuds (appelé temps global). C'est en général la définition qui est choisie pour exprimer ce temps. Le temps de convergence est par essence une variable aléatoire.

Un des buts d'un protocole de population peut être d'obtenir des informations sur l'état initial du graphe en interrogeant un nœud. Dans la section suivante, nous introduisons une classification des protocoles de population, proposant chacun une utilité différente.

Cette classification se fait ici sur base de cinq caractéristiques. La première caractéristique concerne le nombre d'interactions initiées par un nœud, lorsque celui-ci est poussé à interagir. La deuxième est définie sur base de la structure du graphe. La troisième caractéristique est le rythme auquel sont effectuées les interactions entre les nœuds. La quatrième caractéristique est le sens dans lequel l'information est communiquée, lors d'une interaction. La cinquième et dernière caractéristique est le type d'information qui est communiqué, et impacte par conséquent la valeur attendue du nœud, atteinte lorsque l'état de convergence est atteint.

3.2 Caractéristiques des protocoles de population

Dans cette section, nous classifions différents protocoles de population, sur base des caractéristiques introduites à la fin de la section précédente. Cette section a pour but de mieux cerner les différents protocoles existants, leurs différences, et le contexte de leur utilisation.

3.2.1 Nombre d'interactions

Une première caractéristique importante est le nombre d'interactions initiées par un nœud à chaque fois que celui-ci doit interagir.

Comme nous l'avons vu, les nœuds sont amenés à interagir de manière séquentielle. A chaque fois qu'un nœud est amené à agir, il doit initier une ou plusieurs interactions. Ces interactions sont effectuées chacune avec un nœud choisi de manière aléatoire parmi ses voisins. Ce choix est déterminé par la méthode de transmission, abordée dans la seconde caractéristique.

La première caractéristique est donc le nombre d'interactions que chaque nœud doit initier lorsqu'il est amené à interagir. Sur la figure 3.1 est représenté à gauche, le cas où le nombre d'interactions initiées à la fois par un même nœud est de 1. A la i-ème séquence d'interactions ($t = i$), une seule interaction est donc effectuée. Il en est de même à la $(i + 1)$-ème interaction. A droite de cette figure est représenté le cas où le nombre d'interactions effectuées par séquence d'interactions est de deux. Le cercle coloré correspond au nœud qui initie l'interaction, et la flèche en pointillés symbolise l'interaction.

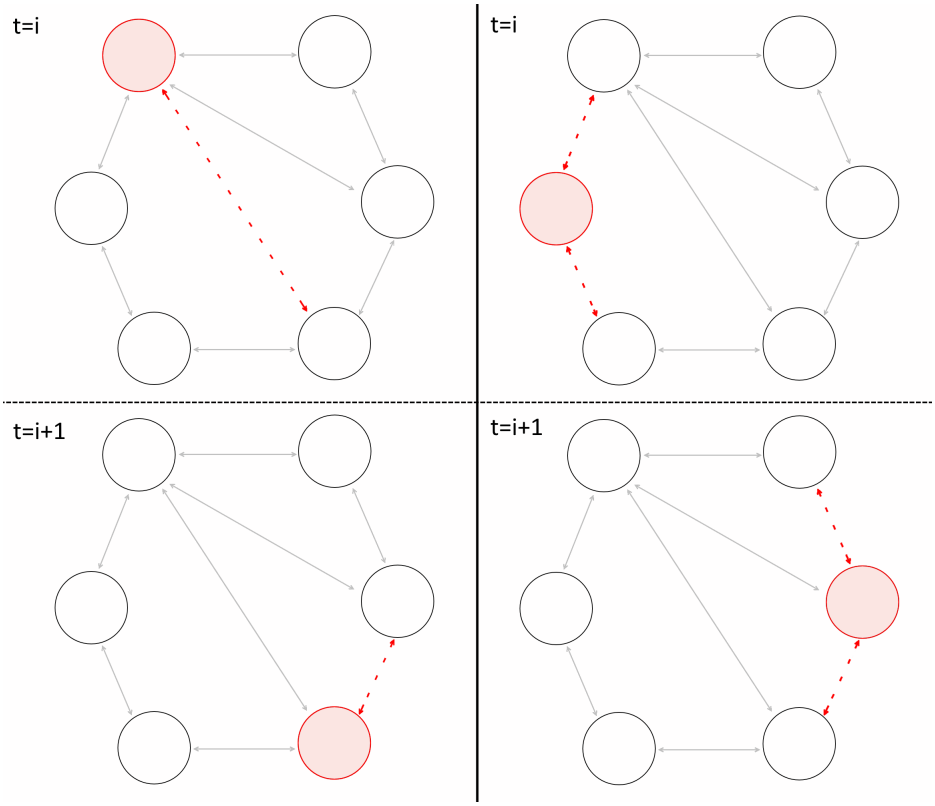

FIGURE 3.1 – Représentations d'une succession d'interactions en fonction du nombre d'interactions simultanées.

3.2.2 Méthode de transmission

Une seconde caractéristique des protocoles de population que nous abordons est la méthode de transmission. Lorsqu'un nœud initie une interaction, il doit déterminer avec quel nœud l'interaction est effectuée. C'est ce choix qui définit la méthode de transmission.

Ce choix peut être effectué de multiples manières, et est entre autre dépendant de la structure du graphe. En effet, dans un graphe où le degré des nœuds (nombre de voisins de chaque nœud) est faible, le choix est beaucoup plus restreint que si tous les nœuds sont voisins. Dans chacune de ces manières, le choix entre les voisins demeure aléatoire. Trois manières de procéder sont introduites dans cette sous-section (figure 3.2).

Une première méthode est de procéder de sorte à ce que le choix du nœud avec lequel l'interaction se produit soit effectué de manière à ce que tous les nœuds du graphe aient la même chance d'être choisi pour interagir. On dit que le choix du nœud est déterminé par une loi uniforme appliquée sur tous les nœuds du graphe. Cela implique que chaque nœud soit voisin avec tous les autres, étant donné que le choix ne peut être effectué qu'entre voisins. On dit alors que le graphe est complet.

La seconde méthode de procéder est de restreindre le nombre de voisins de chaque nœud. Ainsi, un nœud ne pouvant interagir qu'avec ses voisins, le choix du nœud avec lequel interagir suit une loi uniforme appliquée sur un nombre restreint de nœuds du graphe. Cela se traduit par une structure du réseau qui impose que certains capteurs ne soient pas connectés.

Les deux méthodes que nous avons abordées effectuaient leur choix de manière à ce que chaque voisin ait une chance égale d'être choisi. La troisième méthode consiste à ce que les voisins n'aient pas tous une chance égale d'être choisis pour interagir. Cela est modélisé en pondérant les arêtes joignant les nœuds en fonction des probabilités d'observer une interaction entre ces nœuds. Ainsi, le choix n'est plus effectué de manière uniforme, mais serait déterminé en fonction du poids de l'arête rejoignant les nœuds voisins.

Dans la figure 3.2 sont représentées respectivement les trois méthodes décrites ci-dessus.

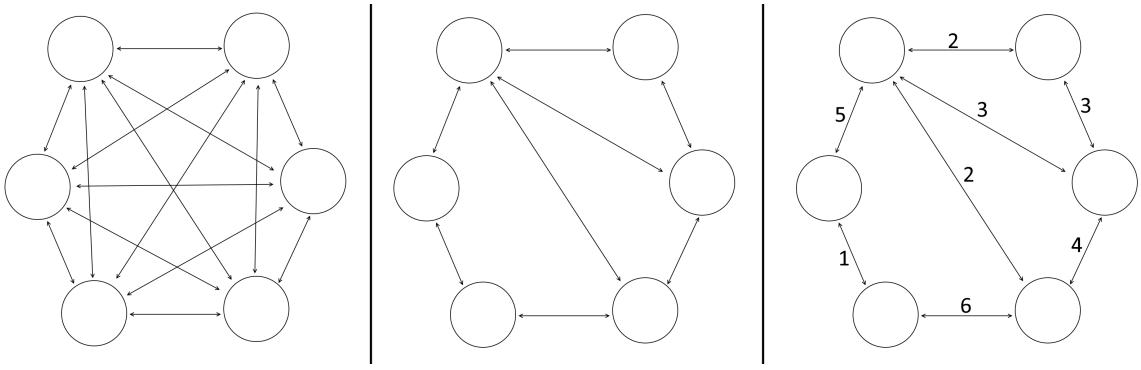

FIGURE 3.2 – Représentation des 3 méthodes

3.2.3 Protocole synchrone/asynchrone

Une troisième caractéristique des protocoles de population détermine le fait que toutes les interactions soient effectuées de manière synchronisée, ou bien soient faites de manière indépendante. Pour aborder cette caractéristique, nous définissons dans cette sous-section le protocole synchrone et asynchrone.

Un protocole de population synchrone est un protocole de population où les interactions prennent place de manière synchronisée. La figure 3.3 représente des vagues d'interactions successives associées à ce caractère synchrone. Le paramètre t est un paramètre sans dimension, exprimant que la t-ème vague d'interactions est concernée. Le cercle coloré correspond au nœud qui initie l'interaction, et la flèche en pointillés symbolise l'interaction. Dans un tel protocole, on considère que tous les nœuds sont associés à une horloge commune (une horloge rythmant les interactions entre les nœuds). Ainsi, les interactions s'exécutent par vagues de manière simultanée pour tous les nœuds. Ces vagues d'interactions prennent donc place successivement. A chaque vague, un certain nombre de nœuds est donc choisi aléatoirement pour initier une interaction avec un autre nœud, cet autre nœud étant choisi en fonction de la méthode de transmission présentée dans la sous-section précédente.

Un protocole de population asynchrone est un protocole où chaque nœud interagit de manière indépendante (non-synchronisée) des autres nœuds. A l'instar de la figure 3.3 pour le protocole synchrone, la figure 3.4 représente des vagues d'interactions pour le protocole asynchrone. Ainsi, chaque nœud a sa propre horloge qui rythme les interactions du nœud en question. L'horloge de chaque nœud détermine de manière aléatoire le temps séparant deux interactions initiées par un nœud. Cet intervalle de temps peut être choisi de plusieurs façons. La manière introduite par Yves Mocquard est de déterminer ce temps à partir d'un processus de Poisson. Cela signifie que le temps entre deux interactions est une variable aléatoire qui suit une loi exponentielle. Dans ce cas, l'horloge associée aux nœuds est appelée horloge exponentielle. Cependant, le comportement de la variable aléatoire peut être défini à l'aide de différentes lois si souhaité. Nous définissons formellement cette horloge dans la section 3.2.5.

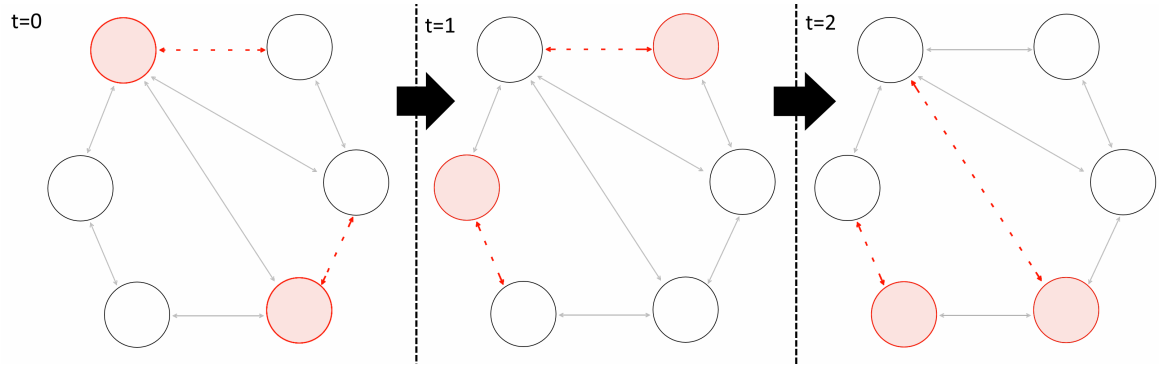

FIGURE 3.3 – Représentation des vagues d'interactions successives associées au mode synchrone.

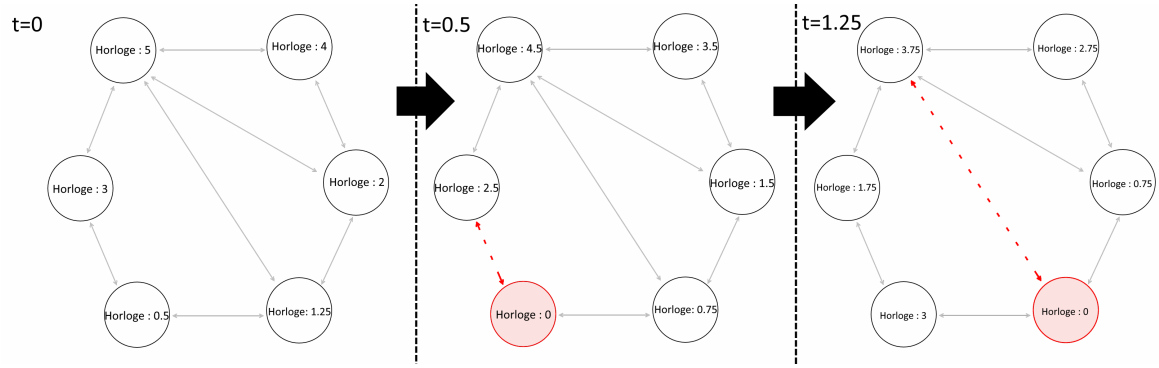

FIGURE 3.4 – Représentation des interactions lors d'un modèle de protocole asynchrone.

3.2.4 Protocole push/pull

Une quatrième caractéristique que nous utilisons afin de classifier les protocoles de population concerne le sens dans lequel l'information est communiquée, lors d'une interaction. Cela est traduit par ce que nous appelons un modèle push, pull, ou push/pull. La figure 3.5 représente ces trois types d'interactions. Le cercle coloré correspond au nœud qui initie l'interaction, et la flèche en pointillés symbolise l'interaction. Le sens de la flèche exprime le sens dans lequel l'information est communiquée. Cette sous-section a pour but de définir ce que sont ces modèles.

Dans un protocole où les interactions sont de type push, le nœud qui a initié une interaction est le seul à envoyer l'information. L'information qu'il connait reste alors invariante lors d'une telle interaction, étant donné qu'aucune mise à jour de l'infor-

mation n'est alors effectuée de son côté.

Dans un protocole où les interactions sont de type pull, au contraire, c'est le nœud qui a initié une interaction qui reçoit l'information, le contraire du modèle push. Ainsi, il est le seul qui est susceptible de voir son information mise à jour.

Finalement, dans le cas où les interactions sont de type à la fois push et pull (noté push/pull), l'information circule dans les deux sens. Le nœud qui initie l'interaction peut avoir son information mise à jour, et le nœud qui subit l'interaction également.

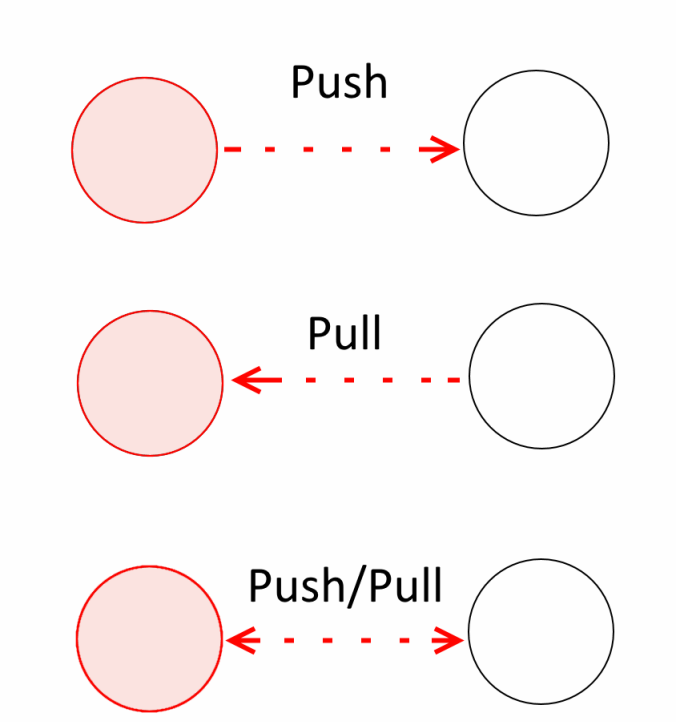

FIGURE 3.5 – Représentation des trois types d'interactions push, pull, et push/pull.

3.2.5 Type d'information

Une cinquième et dernière caractéristique que nous utilisons afin de classifier les différents protocoles de population est le type d'information communiquée entre les nœuds.

Cette caractéristique détermine entre autre l'information qu'un nœud est susceptible

de répondre lorsqu'il est interrogé. Par conséquent, elle détermine ce que l'on souhaite connaître du système initial à l'aide du protocole de population, comme mentionné en section 3.1. C'est donc principalement cette caractéristique qui détermine l'usage que l'on souhaite effectuer du protocole de population.

Cette sous-section a pour but de passer en revue une partie des types d'informations qu'il est possible de trouver dans le cadre des protocoles de population. Nous abordons cinq types d'informations. En premier lieu, le type d'information "minimum/maximum". Dans un second lieu, le type d'information appelé "comptage", suivi par le type "proportion". En quatrième lieu, la "majorité", avant de finir par un type d'information spécial désigné par "Horloge globale". Dans le cadre de cette section, nous considérons que les protocoles sont de type push/pull.

Min/Max

Initialement, on attribue une valeur (par exemple 0 ou 1) à chaque nœud. Le cas "minimum" et "maximum" sont deux types d'informations exclusifs, et sont décrits comme ceci : lorsqu'un nœud interagit avec un autre, les deux nœuds prennent comme valeur la plus grande d'entre les deux nœuds (ou la plus petite, dans le cas "minimum"). Au fur et à mesure des interactions, la valeur associée aux nœuds a donc tendance à augmenter (ou diminuer pour le cas "minimum").

Un graphe associé à ce type d'information a atteint son état de convergence, illustré dans la figure 3.6, lorsque chaque nœud est associé à la valeur maximum parmi les états d'entrée (ou la valeur minimum). Dans la figure 3.6, sont représentés à gauche les états d'entrée des nœuds du graphe. A droite sont affichées les réponses des nœuds lorsque nous souhaitons les interroger une fois l'état de convergence atteint.

Une application de ce type d'information peut se trouver dans la diffusion de rumeur, où on souhaite généralement analyser le temps qu'il faut afin que tout le réseau soit au courant de la valeur maximum parmi les valeurs initiales des nœuds du graphe.

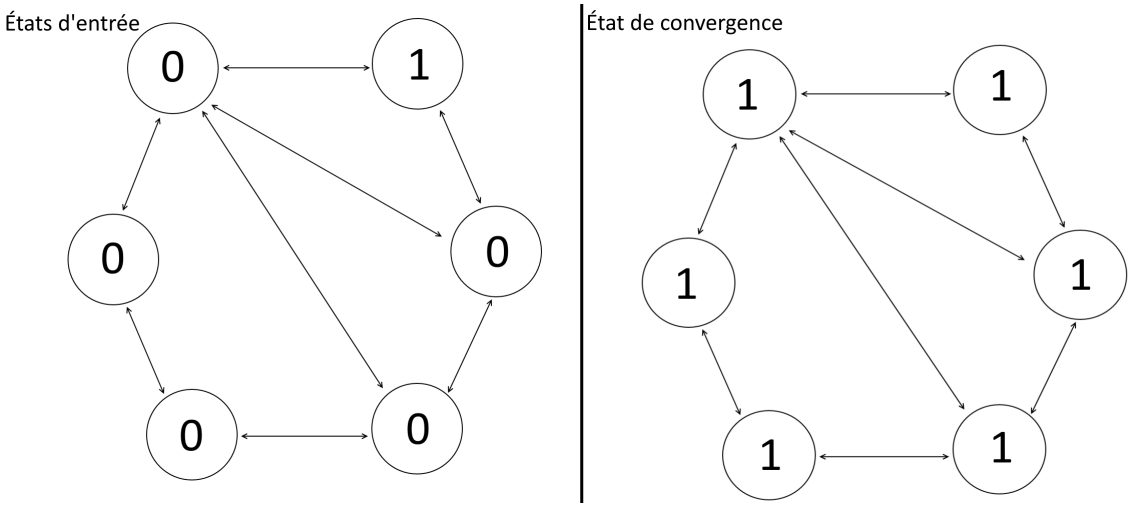

FIGURE 3.6 – Représentation d'un graphe associé à une information de type "maximum".

Comptage

Nous considérons à présent que chaque état d'entrée est déduit de l'ensemble d'états $\{A, B\}$. Le but du comptage est que chaque nœud soit susceptible de connaître le nombre de nœuds initialement associés à l'état A.

Notons n_A le nombre de nœuds associés à l'état d'entrée A. On dit que le graphe a atteint un état de convergence lorsque chaque nœud répond n_A, lorsqu'il est interrogé. Sur la figure 3.7 est représenté un graphe associé à une information de type "comptage". A gauche, sont représentés les états d'entrée des nœuds du graphe. A droite sont affichées les réponses des nœuds lorsque nous souhaitons les interroger une fois l'état de convergence atteint. On remarque que l'état d'entrée n'est pas une valeur numérique, là où la réponse l'est. Il est donc important de préciser que la valeur numérique assignée à un nœud au démarrage est décidée en fonction de son état d'entrée.

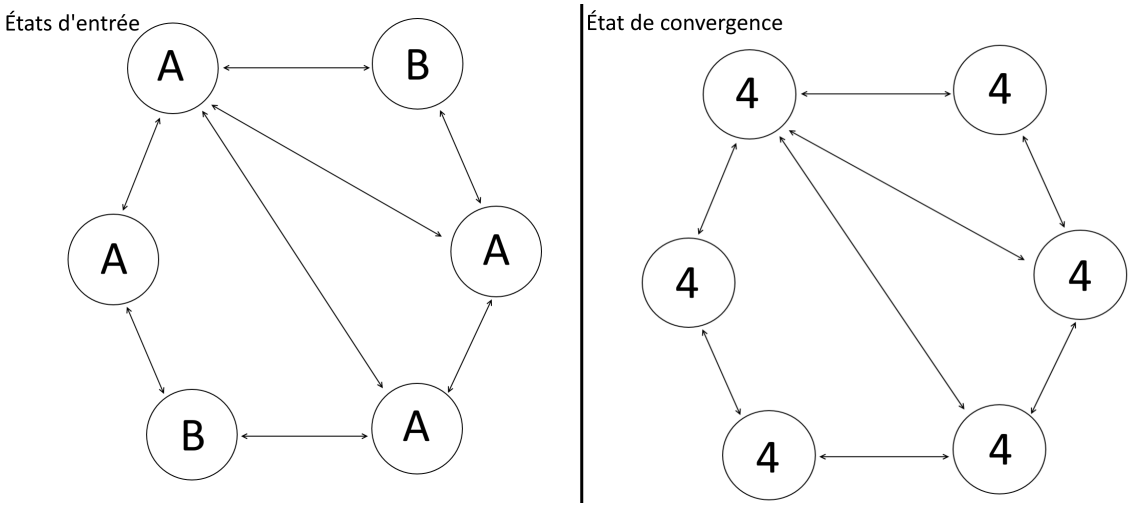

FIGURE 3.7 – Représentation d'un graphe associé à une information de type "comptage".

Proportion

A l'instar du comptage, on considère que chaque nœud démarre en étant associé à un état issu de l'ensemble des états $\{A, B\}$. Le but de ce type d'information est que chaque nœud puisse connaître une valeur approchée de la proportion de nœud ayant débuté avec l'état d'entrée A.

La valeur que l'on souhaite approcher afin de garantir l'état de convergence est ici envisagée avec une certaine précision, notée ϵ. A nouveau, notons n_A le nombre de nœuds associés à l'état d'entrée A. Ainsi, l'état de convergence est atteint lorsque chaque nœud du graphe répond la proportion de nœuds associés à l'état d'entrée A, à ϵ près. Autrement dit, lorsque pour tous les nœuds, $|reponse - \frac{n_A}{n}| \leq \epsilon$.

Par exemple, considérons un ensemble de 6 nœuds, où 4 de ces nœuds commencent dans l'état A et 2 dans l'état B. La proportion de nœuds commençant dans l'état A est donc de 2/3. Dans l'état de convergence, chaque nœud doit donc être égal à 2/3 avec une marge d'erreur de ϵ. Fixons ici ϵ à $\frac{0.15}{3}$. La figure 3.8 représente ce cas. A gauche, sont représentés les états d'entrée des nœuds du graphe. A droite sont affichées les réponses des nœuds lorsque nous souhaitons les interroger une fois l'état de convergence atteint. Les valeurs associées à l'état de convergence sont ici atteintes avec une précision de ϵ, qui est ici d'une valeur égale à 0.15/3.

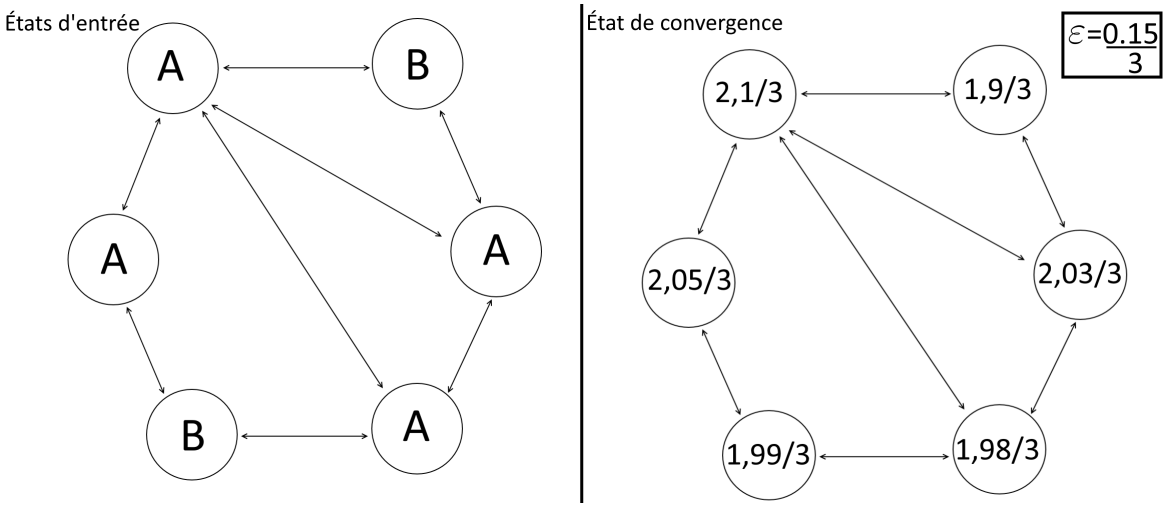

FIGURE 3.8 – Représentation d'un graphe associé à une information de type "proportion".

Majorité

Considérons à nouveau chaque état d'entrée comme appartenant à l'ensemble d'états $\{A, B\}$. La majorité consiste à déterminer si l'état A est majoritaire parmi les états d'entrée.

Si n_A est plus grand que n_B, la valeur que l'on souhaite observer dans les nœuds est 1 (n_A étant défini précédemment, et n_B étant défini de manière similaire à n_A). Sinon, un nœud devrait renvoyer 0. L'état de convergence du graphe est atteint lorsque chaque nœud répond "1" si $n_A > n_B$, ou chaque nœud répond "0" dans le cas contraire. La figure 3.9 correspond à un graphe associé à une information de type "majorité". A gauche, sont représentés les états d'entrée des nœuds du graphe. A droite sont affichées les réponses des nœuds lorsque nous souhaitons les interroger une fois l'état de convergence atteint. Comme $n_A > n_B$, l'état de convergence est celui où chaque nœud est associé à la valeur "1".

L'égalité $n_A = n_B$ n'est pas considérée, et est envisagée comme associée à un protocole qui ne converge pas.

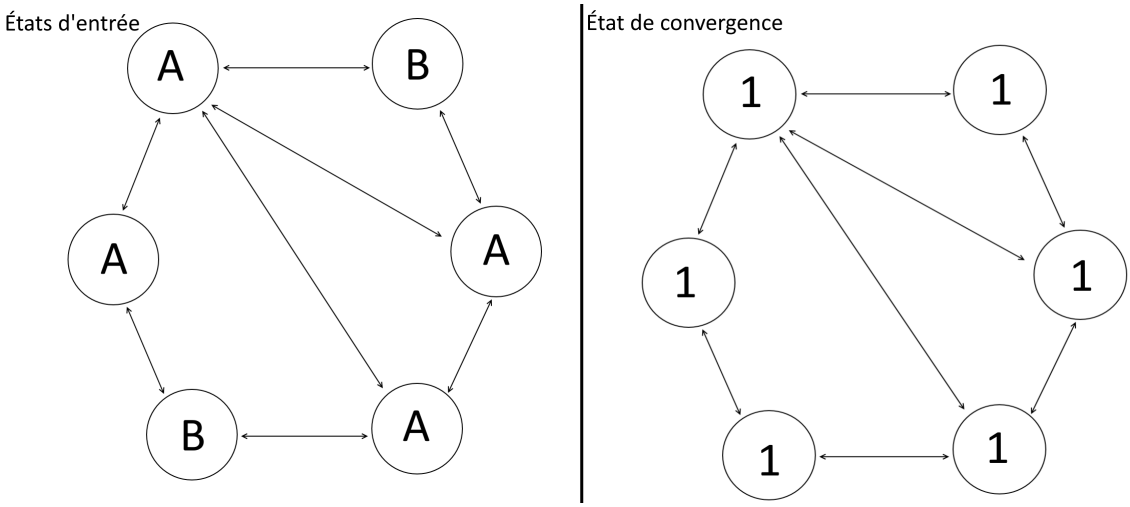

FIGURE 3.9 – Représentation d'un graphe associé à une information de type "majorité".

Horloge globale

L'horloge globale est un cas très particulier dans la mesure où il implique que le protocole ne converge pas. Afin d'introduire l'horloge globale, il faut d'abord définir la notion de temps global et de temps parallèle global. A la différence des trois protocoles précédents, ce protocole ne cherche pas à donner une information sur l'état initial du graphe.

Le temps global, noté t, est le nombre d'interactions qui ont été initialisées depuis le début jusqu'au moment où nous observons le système. Le temps parallèle global, lui, correspond au nombre moyen d'interactions initialisées par un nœud depuis le début. Ainsi, ce temps parallèle global n'est ni plus ni moins que le nombre d'interactions ayant eu lieu depuis le démarrage divisé par la taille du réseau, soit t/n. C'est en quelque sorte une manière de garder une certaine notion du temps plus réaliste pour chaque nœud.

L'objectif de l'horloge globale est que chaque nœud soit capable de calculer une estimation du temps parallèle global t/n avec une certaine précision. Ce protocole est souvent couplé aux autres afin que chaque nœud puisse garder une idée du temps qui s'écoule.

3.3 Contraintes d'implémentation

Cette section a pour but d'aborder brièvement les quelques contraintes majeures qui peuvent intervenir lors de l'implémentation de protocoles de population, pouvant dès lors motiver l'utilisation de certains protocoles. Nous abordons ici deux contraintes : la contrainte d'espace de mémoire et la contrainte de durée de vie de l'information.

Une première contrainte qui peut survenir est le fait que la mémoire des différents composants d'un réseau sur lequel un tel protocole doit être mis en place soit limitée. Bien que cela ne puisse pas sembler conséquent dans des cas où nous appliquons un protocole de population en utilisant un nombre d'états très limité, adopter un nombre élevé d'états pourrait rendre la charge de mémoire importante, impliquant la nécessité d'une gestion de l'information plus adéquate.

Cela est d'autant plus vrai que certains états peuvent à eux seuls représenter une masse de données considérable. Si nous reprenons la première application abordée au chapitre 2 [4], concernant la réplication des changements dans des bases de données à l'aide d'un mécanisme de diffusion de rumeur, cela peut se montrer problématique. Effectivement, si la base de données représente une masse d'informations trop élevée, il peut être avisé de se rabattre sur des solutions moins gourmandes en mémoire.

Une seconde contrainte qui peut être prise en compte dans le cadre de ces protocoles est la durée de vie que peut avoir une information. D'ailleurs, dans les premières parutions des protocoles de diffusion de rumeur en 1989, dans le cadre informatique, on considère que l'information a une durée de vie limitée [4], assurant par conséquent une certaine probabilité que l'information n'ait pas eu le temps d'être communiquée à travers tout le réseau.

Ces différentes contraintes peuvent dès lors être une source de motivation pour envisager une modification sur le type de données traitées, ou sur le protocole employé.

3.4 Conclusion

Dans la suite de ce mémoire, nous nous restreindrons à l'analyse d'un cas en particulier : la diffusion de rumeur. Nous abordons dès lors à présent comment se classifie ce protocole parmi les différentes caractéristiques précédemment introduites.

Nous introduisons et analysons un protocole push/pull, utilisant un type d'information min/max (ici maximum). Nous effectuons une première analyse à l'aide du modèle synchrone, pour ensuite décrire le modèle asynchrone dans une autre section. Nous traduisons la connaissance de la rumeur par un état "1" et l'ignorance de cette rumeur par un état "0". Nous limitons le nombre d'interactions initiées à la fois par un nœud à un seul autre nœud voisin, choisi de manière uniforme parmi tous les autres nœuds du graphe (en considérant dès lors un graphe complet). Le processus stochastique qui régit les instants où une interaction a lieu est le processus de Poisson.

Ainsi, chaque nœud a une horloge indépendante rythmant ses interactions, à l'issue desquelles le nœud est susceptible d'apprendre ou de propager la rumeur.

Chapitre 4

Protocole de diffusion de rumeur

4.1 Introduction

Dans ce chapitre, nous décrivons le modèle régissant un protocole de diffusion de rumeur. Comme mentionné dans le chapitre précédent, le protocole de diffusion de rumeur abordé est un protocole push/pull où l'information est de type min/max. La structure du graphe sur lequel nous travaillons est telle que le graphe est complet. Chaque nœud a la même chance d'être choisi pour participer à une interaction.

Ce chapitre est décomposé en trois sections. La première concerne le problème de diffusion de rumeur en temps discret, à l'aide du modèle synchrone. La seconde section concerne l'étude du temps de convergence T_n et de sa distribution. Finalement, la troisième section aborde le protocole dans sa forme asynchrone, en employant un temps continu pour faire évoluer le graphe.

La première et troisième sections ont pour but principal de décrire de manière intuitive le fonctionnement du protocole, afin de fournir les bases suffisantes pour une éventuelle implémentation. A ce but vient se greffer une brève analyse du temps de convergence du protocole concerné, sur base des résultats de la thèse d'Yves Mocquard.

4.2 Cas en temps discret

Cette section est axée autour du protocole de diffusion de rumeur synchrone. Dans un premier temps, nous regardons comment modéliser le protocole. Dans un second

temps, nous analysons le comportement de la diffusion d'une rumeur sur le graphe, à l'aide d'une approche Markovienne. Dans cette approche nous modélisons le protocole sous la forme d'une collection d'états, où une transition entre états peut être opérée sur base d'une certaine probabilité. Dans un troisième et dernier temps, nous analysons expérimentalement le temps de convergence que nous pouvons espérer, en fonction de la taille du graphe concerné.

4.2.1 Réseau et états des différents nœuds

Comme nous l'avons vu au cours du chapitre 3, nous modélisons le réseau sous forme d'un graphe. Ce graphe est composé de n nœuds. Chaque nœud est susceptible de connaître une information. En l'occurrence, l'information est ici une rumeur. Une valeur représentant cette information est associée à chaque nœud. Par souci de simplicité, on considère qu'un nœud dont la valeur associée est 0 ne connait pas la rumeur. En revanche, si sa valeur associée est 1, il connait alors la rumeur.

L'état d'entrée d'un nœud est pour rappel l'état déterminant la valeur initiale de ce nœud. Cet état est donc compris dans l'ensemble $\{0, 1\}$. La valeur renvoyée par le nœud lorsque nous souhaitons l'interroger est également compris dans l'ensemble $\{0, 1\}$. Lorsque nous interrogeons un nœud, il nous renvoie "1" s'il connait la rumeur, et "0" dans le cas contraire.

Lors d'une interaction, si un des deux nœuds qui interagissent connait la rumeur, les deux connaissent la rumeur à l'issue de l'interaction. Autrement dit, nous pouvons représenter la fonction régissant l'interaction de la manière suivante :

$$\begin{aligned} f := \{0,1\}\times\{0,1\} &\to \{0,1\} \times \{0,1\} \\ (x, y) &\to (max\{x, y\}, max\{x, y\}) \end{aligned} \tag{4.1}$$

où (x, y) est donc le couple formé par les valeurs des deux nœuds qui interagissent, leur nouvelle valeur associée étant déterminée par la fonction $f(x, y)$.

De plus, comme mentionné dans le chapitre précédent, nous utilisons une notion du temps t qui est défini comme le nombre d'interactions ayant eu lieu depuis l'initialisation du graphe. Par conséquent, le temps de convergence T_n est exprimé comme le nombre d'interactions nécessaires pour que les n nœuds connaissent la rumeur. Ce

temps de convergence est noté T_n.

A chaque instant t, la configuration du protocole est notée C_t. Cette configuration est un vecteur ligne de taille n. Le i-ème élément de ce vecteur correspond à l'état du i-ème du nœud du graphe, avec $i \in \{1, \ldots, n\}$. La configuration C_t est donc définie comme ceci :

$$C_t = (C_t^{(1)}, \ldots C_t^{(n)}) \tag{4.2}$$

et où $C_t^{(i)}$ est la valeur associée au i-ème nœud à l'instant t.

4.2.2 Protocole de diffusion de la rumeur

Regardons à présent comment se comporte exactement le protocole, étape par étape, afin de pouvoir ultérieurement en modéliser le comportement à l'aide d'une chaîne de Markov.

Pour pouvoir propager une rumeur, il est nécessaire qu'au moins un nœud connaisse la rumeur, initialement. Dès lors, le nombre de nœuds initialement associés à la valeur "1" est strictement positif. Une fois les valeurs associées aux différents nœuds initialisées, nous pouvons commencer à les faire interagir entre eux.

Tour à tour, nous choisissons deux nœuds aléatoirement afin que ceux-là interagissent entre eux. Nous notons les indices de ces deux nœuds respectivement i et j, avec la contrainte $1 \leq i, j \leq n$ et $i \neq j$. Le choix de i est effectué de manière uniforme parmi tous les nœuds, ce qui signifie que chaque nœud a une probabilité $1/n$ d'être choisi. Le graphe étant complet, et j devant être différent de i, le choix de j est également effectué parmi tous les autres nœuds que le nœud i. Ainsi, chaque autre nœud a une probabilité $\frac{1}{n-1}$ d'être choisi pour interagir avec le i-ème nœud.

Notons désormais X_t la variable aléatoire qui correspond au choix des indices i, j à l'instant t. Si ce choix est uniforme (ce qui est le cas dans notre situation), nous pouvons écrire cette probabilité ainsi :

$$\mathbb{P}\{X_t = (i,j)\} = \frac{1}{n(n-1)} 1_{i \neq j}, \qquad \forall i, j \in [\![1, n]\!] \tag{4.3}$$

où, pour rappel, $1_{i \neq j}$ est égal à 1 tant que $i \neq j$, et 0 dans le cas contraire.

Une fois un couple (i, j) trouvé, l'interaction prend place entre les deux nœuds. Cette interaction est déterminée par la fonction définie en (4.1). Lors de l'interaction, les valeurs des deux nœuds sont communiquées dans les deux sens, comme le veut le type d'interaction push/pull. Comme le type d'information transmise est de type "maximum" (sous-section 3.2.5), la nouvelle valeur associée aux i-ème ($C_t^{(i)}$) et j-ème ($C_t^{(j)}$) nœuds à l'instant t est défini comme ceci :

$$C_t^{(i)} = C_t^{(j)} = \max\{C_{t-1}^{(i)}, C_{t-1}^{(i)}\} \tag{4.4}$$

Les autres nœuds qui n'interagissent pas préservent les valeurs qui leur étaient associées en $t-1$. Par conséquent,

$$C_t^{(r)} = C_{t-1}^{(r)}, \qquad r \neq i, j \tag{4.5}$$

Ainsi, à chaque instant t, un nouveau couple de nœuds est choisi aléatoirement afin d'interagir, et les valeurs des nœuds du graphe évoluent de manière telle que nous venons de l'aborder. Les interactions peuvent éventuellement prendre fin lorsque l'état de convergence est atteint. Cet état est atteint lorsque la valeur associée à chaque nœud est égale à 1. Cela correspond à l'état où tous les nœuds du réseau connaissent la rumeur tel que :

$$C_t^{(i)} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{4.6}$$

Comme vu précédemment, le temps écoulé jusqu'à atteindre l'état de convergence est noté T_n. Ce temps T_n peut être défini mathématiquement de la manière suivante :

$$T_n = \inf\left(t \geq 0 \left| \sum_{i=1}^{n} C_t^{(i)} = n \right.\right) \tag{4.7}$$

Une représentation de l'évolution du réseau peut être effectuée telle que le propose la figure 4.1. L'évolution du réseau est déterminée à l'aide d'un protocole de diffusion de rumeur en temps discret, avec un seul nœud connaissant initialement la rumeur. En pointillés est représentée l'interaction prenant place à l'instant t.

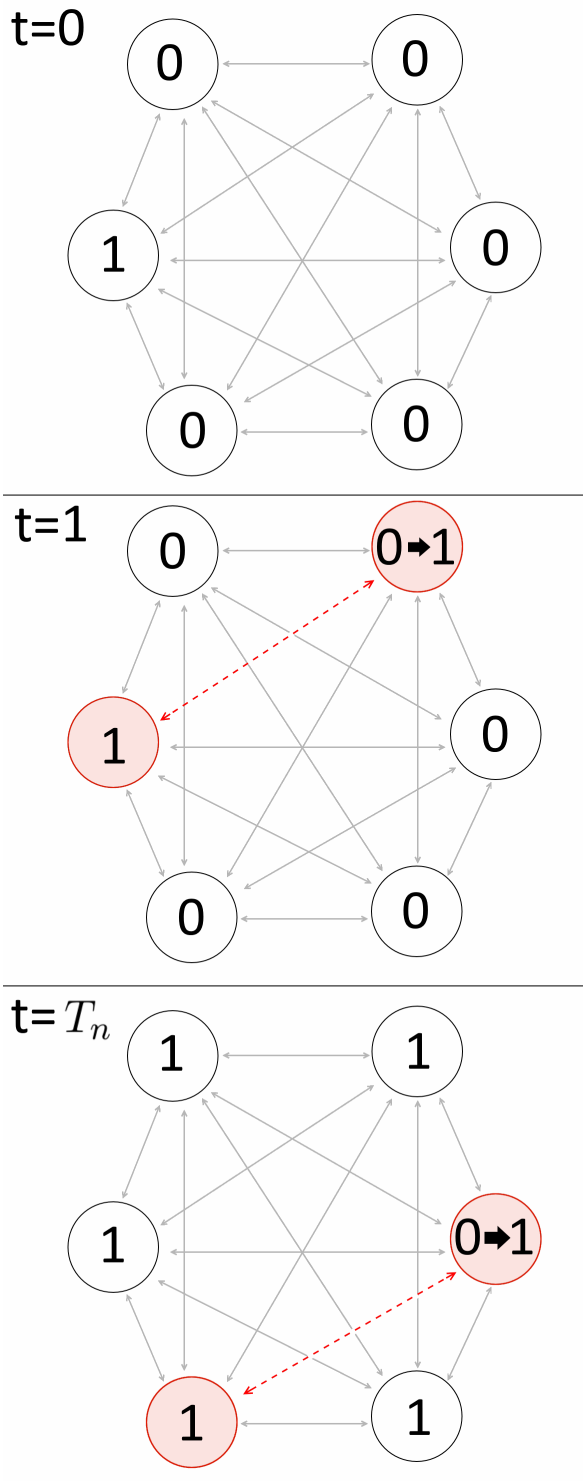

FIGURE 4.1 – Évolution d'un réseau de taille $n = 6$ en temps discret.

4.2.3 Approche Markovienne

Dans le cadre des prochains développements, nous représentons le réseau sous forme d'un processus stochastique. Une fois cela effectué, nous montrerons que ce processus stochastique est une chaîne de Markov. Cela nous permet de proposer une distribution du temps de convergence.

Dans la suite de ce travail, nous manipulons des processus stochastiques à temps discret. En voici la définition :

Définition - Processus stochastique à temps discret

Un processus stochastique à temps discret est une famille $X = \{X_n, n \in \mathbb{N}\}$ de variables aléatoires.

Introduisons à présent le processus stochastique à temps discret $Y = \{Y_t, t \in \mathbb{N}\}$ qui nous intéresse dans notre étude. Par définition, Y_t est une variable aléatoire. La variable Y_t correspond dans notre cas au nombre de nœuds connaissant la rumeur à l'instant t. Elle est définie telle que

$$Y_t := \sum_{i=1}^{n} C_t^{(i)} \tag{4.8}$$

où pour rappel, $C_t^{(i)}$ est la valeur associée au i-ème nœud à l'instant t.

L'évolution de notre réseau au fil du temps peut être représentée à l'aide du processus stochastique à temps discret Y que nous venons d'introduire. Effectivement, Y_t se suffit à lui seul pour connaître l'état du réseau à tout instant $t \in \mathbb{N}$.

Nous allons à présent établir que Y_t est un processus stochastique particulier, à savoir une chaîne de Markov. En voici la définition en toute généralité :

Propriétés - Chaîne de Markov

Soit un espace d'états E dénombrable (sans perte de généralité, posons $E = \{1, 2, \ldots, i\}$ avec i fini). Une chaîne de Markov à temps discret $X = \{X_n, n \in \mathbb{N}\}$ à valeur dans E est un processus stochastique à temps discret vérifiant les deux propriétés suivantes :

— $\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$ $(\star)$

— $\mathbb{P}(X_{n+1} = y | X_n = x)$ est indépendant dz n.

pour tout $n \geq 0$ et pour tout $x_i \in E$ où $i \in \{1, \ldots, n\}$.

De plus, si X est une chaîne de Markov à valeur à temps discret dans E, alors il existe des probabilités $P(x, y)$ avec $x, y \in E$ appelées probabilités de transition définies telles que pour tout $n \in \mathbb{N}$:

$$P(x_n, x_{n+1}) := \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n) \tag{4.9}$$

Ces probabilités $(P(x, y))_{x,y \in E}$ peuvent être exprimées sous forme d'une matrice appelée matrice de probabilité de transition. Les éléments de cette matrice sont notés $P_{x,y}$, et correspondent à la probabilité de transiter d'un état x à un état y.

Sur base de la propriété $(\star)$, on observe que la variable X_{n+1} est indépendante des variables X_i où $i < n$. Cette variable X_{n+1} est donc déterminée uniquement à partir de la valeur de X_n. La première propriété citée est également appelée propriété de Markov.

Dans notre cas, le processus stochastique à temps discret Y est bien une chaîne de Markov :

— seul le nombre Y_t de nœuds connaissant la rumeur à l'instant t détermine la probabilité que Y_{t+1} nœuds connaissent la rumeur à l'issue de la prochaine interaction

— la valeur de t n'a aucune influence sur le comportement de notre réseau, vérifiant ainsi la seconde propriété.

Le processus Y est donc une chaîne de Markov à n états. Les n états correspondent aux n valeurs pouvant être prises par la variable aléatoire Y_t. Cette dernière est un naturel compris entre 1 et n.

Le processus Y étant une chaîne de Markov, une matrice de transition peut lui être associée. Notons cette matrice de transition A, définie comme ceci, pour $i \in \{1, \ldots, n\}$:

$$A_{i,i} := \mathbb{P}(Y_{n+1} = i | Y_n = i) = 1 - \frac{2i(n-i)}{n(n-1)} \tag{4.10}$$

$$A_{i,i+1} := \mathbb{P}(Y_{n+1} = i+1 | Y_n = i) = \frac{2i(n-i)}{n(n-1)}, \qquad i \neq n \tag{4.11}$$

où $A_{i,i}$ correspond à la probabilité de rester dans l'état i suite à une interaction, c'est-à-dire la probabilité que la rumeur ne soit pas propagée davantage à l'issue de l'interaction. La probabilité $A_{i,i+1}$ correspond à la probabilité que la rumeur soit propagée, et que le nombre de nœuds connaissant la rumeur passe de i à $i+1$. Cette dernière probabilité est positive jusqu'à ce qu'on atteigne $i = n$, ce qui correspond à l'état de convergence. Toutes les autres probabilités sont égales à 0, dans la matrice A, étant donné que le nombre de nœuds connaissant la rumeur ne peut pas décroître, et étant donné qu'on ne peut atteindre l'état $i+1$ que depuis l'état i ou l'état $i+1$.

Développons à présent l'expression (4.11). La propagation de la rumeur résulte de l'interaction d'un nœud connaissant la rumeur et d'un nœud ne la connaissant pas. Soit deux nœuds choisis aléatoirement pour interagir. Deux cas peuvent expliquer la propagation de la rumeur. Soit le premier des deux nœuds connait la rumeur et le deuxième ne la connait pas, soit c'est le deuxième nœud qui connait la rumeur et le premier nœud ne la connait pas.

Pour le premier cas, la probabilité que le premier des deux nœuds connaisse la rumeur est de i/n, et la probabilité que le second nœud ne la connaisse pas est de $(n-i)/(n-1)$, c'est-à-dire le nombre de nœuds ne connaissant pas la rumeur divisé par le nombre de nœuds restants (le premier nœud ayant déjà été choisi). Ainsi, la probabilité que le premier cas prenne place lors d'une interaction est :

$$\frac{i}{n} \times \frac{n-i}{n-1} \tag{4.12}$$

Pour le second cas, la probabilité pour que ce soit le premier nœud qui ne connaisse pas la rumeur est de $(n-i)/n$, et que le second nœud connaisse la rumeur est de $i/(n-1)$. La probabilité que ce soit ce deuxième cas qui prenne place lors d'une interaction est :

$$\frac{n-i}{n} \times \frac{i}{n-1} \tag{4.13}$$

La probabilité qu'un nœud apprenne la rumeur à la suite d'une interaction est la probabilité que soit le premier cas prenne place, plus la probabilité que le second cas prenne place. Cette probabilité peut être exprimée à partir des expressions

$$A_{i,i+1} = \frac{i}{n} \times \frac{n-i}{n-1} + \frac{n-i}{n} \times \frac{i}{n-1} = \frac{2i(n-i)}{n(n-1)} \tag{4.14}$$

Assez logiquement, la seule issue possible d'une interaction, outre le fait qu'un nœud ait appris la rumeur, est qu'aucun nœud n'ait appris la rumeur. Dès lors, la probabilité $A_{i,i}$ de rester dans l'état $Y_t = i$ est de 1 moins la probabilité $A_{i,i+1}$, c'est-à-dire

$$A_{i,i+1} = \frac{2i(n-i)}{n(n-1)}, \qquad i \neq n \tag{4.15}$$

Notons p_i la probabilité $A_{i,i+1}$, pour $i \in \{1, \ldots, n-1\}$ telle que

$$p_i := \mathbb{P}\{Y_{t+1} = i+1 | Y_t = i\} = \frac{2i(n-i)}{n(n-1)} \tag{4.16}$$

Nous pouvons dès lors représenter l'évolution de la chaîne de Markov Y à l'aide d'un graphe appelé graphe de transition, présenté dans la figure 4.2 à l'aide des notations introduites précédemment. Cette figure est issue du chapitre 4 de la thèse d'Yves Mocquard [11].

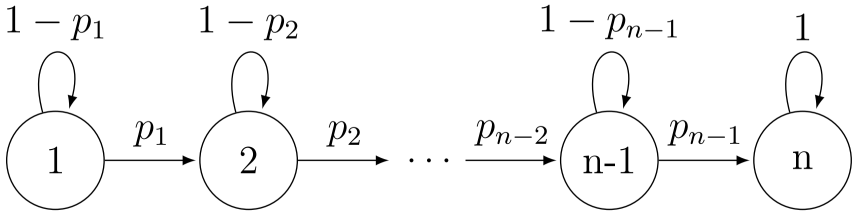

FIGURE 4.2 – Graphe de transitions de la chaîne de Markov Y.

Le temps de convergence T_n peut désormais être écrit de la manière suivante :

$$T_n = \inf\{t \geq 0 | Y_t = n\} \tag{4.17}$$

soit le temps minimum pour que le nombre de nœuds connaissant la rumeur soit égal à n.

Dans la suite de cette section, nous voulons obtenir la distribution du temps de convergence T_n afin d'obtenir une expression facilement calculable de cette dernière. Pour se faire, nous partons de l'équation (4.18) pour obtenir une expression de récurrence. Les preuves concernant les différentes expressions ne sont pas développées dans le cadre de ce mémoire, mais demeurent accessibles dans le chapitre 4 de la thèse d'Yves Mocquard [11].

Pratiquement, la distribution de T_n est donnée par

$$\mathbb{P}\{T_n > k\} = \alpha Q^k \mathbb{1} \tag{4.18}$$

où α est le vecteur ligne défini par

$$\alpha_i := \mathbb{P}\{Y_0 = i\} = 1_{\{i=1\}}, \tag{4.19}$$

La matrice Q est quand à elle la sous-matrice de A où sont supprimées la ligne et la colonne qui correspondent à l'état absorbant n, et où $\mathbb{1}$ est un vecteur colonne de dimension $n-1$ dont toutes les composantes sont égales à 1.

Ce résultat est un résultat connu dans la théorie des chaînes de Markov. Il concerne le temps de séjour dans une chaîne de Markov. En effet, T_n peut être vu comme le temps de séjour du processus Y dans les états $\{1, \ldots, n-1\}$ avant absorption dans l'état n.

Explicitons à présent l'expression (4.18). Afin d'obtenir une expression récursive de la distribution de T_n, nous développons d'abord une expression de récurrence définie comme suit. Soit $V(k) = (V_1(k), \ldots, V_{n-1}(k))$ le vecteur colonne défini par $V_i(k) = \mathbb{P}\{T_n > k | Y_0 = i\}$. Nous avons alors :

$$\begin{aligned} V_i(k) &= (1-p_i)V_i(k-1) + p_i V_{i+1}(k-1) \qquad i \in \{1, \ldots, n-2\} \\ V_{n-1}(k) &= (1-p_{n-1})V_{n-1}(k-1) \end{aligned} \tag{4.20}$$

avec $V(0) = \mathbb{1}$ et où pour rappel $p_i = 2i(n-i)/(n(n-1))$. Ces relations de récurrence sont obtenues rapidement lorsqu'on utilise la physique du système.

Ces équations de récurrence peuvent être utilisées afin d'obtenir une expression explicite pour $V_{n-1}(k)$:

$$\mathbb{P}\{T_n > k | Y_0 = n-1\} = \sum_{j=1}^{\lfloor n/2 \rfloor} (c_{i,j}(1-p_j) + kd_{i,j})(1-p_j)^{k-1} \tag{4.21}$$

où les coefficients $c_{i,j}$ et $d_{i,j}$ sont donnés par $c_{1,j} = 1_{\{j,1\}}$ et $d_{1,j} = 0$ pour $j \in \{1, \ldots, n-1\}$. Pour $(i,j) \in \{2, \ldots, n-1\} \times \{1, \ldots, n-1\}$, nous avons

$$\begin{aligned}
c_{i,j} &= \frac{p_i c_{i-1,j}}{p_i - p_j} - \frac{p_i d_{i-1,j}}{(p_i - p_j)^2} \\
d_{i,j} &= \frac{p_i d_{i-1,j}}{p_i - p_j} \\
c_{i,i} &= 1 - \sum_{j=1, j\neq i}^{\lfloor n/2 \rfloor} c_{i,j} \\
c_{i,n-i} &= 1 - \sum_{j=1, j\neq n-i}^{\lfloor n/2 \rfloor} c_{i,j} \\
d_{i,i} &= p_i c_{i-1,i} \\
d_{i,n-i} &= p_i c_{i-1,n-i}
\end{aligned} \tag{4.22}$$

Les démonstrations concernant les différentes étapes liant ces dernières expressions étant relativement lourdes et le mémoire n'ayant pas pour but de s'attarder sur les détails mathématiques des preuves des différents théorèmes, nous laissons à la discrétion du lecteur le choix d'aller lire les détails des différentes preuves dans la thèse d'Yves Mocquard [11].

Lorsque n devient trop grand, les opérations requises afin d'obtenir les différents coefficients pour le calcul de T_n deviennent trop lourdes et impliquent l'utilisation d'une alternative afin de borner le temps de convergence. C'est dans cette optique que la prochaine section est développée.

4.3 Bornes de la distribution de T_n

Dans la section précédente, nous avons vu une manière d'exprimer explicitement la distribution de T_n. Cependant, la complexité de l'expression ainsi que le temps requis

pour le calcul des coefficients pouvant être problématiques lorsque n est trop grand, nous proposons dans cette section une manière d'estimer un majorant du temps de convergence T_n.

Dans ce contexte, l'équation de récurrence (4.20) peut être développée afin d'obtenir l'expression suivante pour $n \geq 2$ et $k \geq 1$:

$$\mathbb{P}\{T_n > k | Y_0 = 1\} \leq \left(1 + \frac{2k(n-2)^2}{n}\right)\left(1 - \frac{2}{n}\right)^{k-1} \tag{4.23}$$

Il peut être pertinent de noter que lorsque $k \to \infty$, cette inégalité devient une approximation.

Pour la suite de cette section, il est nécessaire de connaitre l'espérance de T_n, notée $\mathbb{E}(T_n)$. Comme nous l'avons vu, elle est initialement donnée par

$$\mathbb{E}(T_n) = \alpha(I - Q)^{-1}\mathbb{1} \tag{4.24}$$

où pour rappel, α, Q et $\mathbb{1}$ sont définis dans l'expression (4.19). Cette espérance peut également s'écrire de la manière suivante :

$$\mathbb{E}(T_n | Y_0 = i) = \frac{(n-1)(H_{n-1} + H_{n-i} - H_{i-1})}{2} \tag{4.25}$$

où H_k est la k-ième somme partielle de la série harmonique, qui est définie ainsi :

$$\begin{aligned} H_0 &:= 0 \\ H_k &:= \sum_{l=1}^{k} \frac{1}{l} \qquad k \geq 1 \end{aligned} \tag{4.26}$$

Une fois de plus, la preuve de l'expression (4.25) n'est pas développée dans le cadre de ce mémoire. Développons à présent la borne de la distribution de T_n.

Soit c un nombre réel, et $\mathbb{E}(T_n)$ l'espérance du temps de convergence. En appliquant $k = \lfloor c\mathbb{E}(T_n) \rfloor$ à (4.23), nous obtenons

$$\begin{aligned}\mathbb{P}\{T_n > c\mathbb{E}(T_n)\} &\leq \left(1 + \frac{2\lfloor c\mathbb{E}(T_n)\rfloor (n-2)^2}{n}\right)\left(1 - \frac{2}{n}\right)^{\lfloor c\mathbb{E}(T_n)\rfloor - 1} \\ &\leq \left(1 + \frac{2c\mathbb{E}(T_n)(n-2)^2}{n}\right)\left(1 - \frac{2}{n}\right)^{c\mathbb{E}(T_n)-2} \qquad (4.27)\end{aligned}$$

Cette expression est notée $f(c, n)$. Intuitivement, l'usage de c détermine dans quelle portion de la queue de la distribution de T_n nous souhaitons faire porter la borne. Par exemple, un c égal à 1 signifie que nous voulons connaître la probabilité que le temps de convergence soit supérieur à l'espérance de ce dernier. Un c très grand traduit le fait que nous souhaitons connaître la probabilité qu'il faille attendre très longtemps avant que tous les nœuds ne connaissent la rumeur.

En remplaçant l'espérance par son expression (5.3), nous obtenons

$$f(c, n) = \left(1 + \frac{2c(n-1)H_{n-1}(n-2)^2}{n}\right)\left(1 - \frac{2}{n}\right)^{c(n-1)H_{n-1}-2} \qquad (4.28)$$

Analysons à présent le comportement asymptotique de cette borne $f(c, n)$.

Pour tout nombre réel $c > 1$, lorsque n tend vers l'infini, il peut être prouvé que la borne $f(c, n)$ tend vers 0. La preuve se base sur la propriété suivante liée aux séries harmoniques : $ln(n) \leq H_{n-1} \leq 1 + ln(n-1)$.

Si $c < 1$, il peut être prouvé qu'au contraire, $f(c, n)$ tend vers 1 lorsque n tend vers l'infini.

Finalement, pour $c = 1$, nous pouvons trouver

$$\lim_{n\to\infty} \mathbb{P}\{T_n > \mathbb{E}(T_n)\} = 0.448429663727 \qquad (4.29)$$

Autrement dit, cela traduit que pour une taille de réseau tendant vers l'infini, la probabilité que le temps de convergence soit supérieur à l'espérance de ce même temps

est d'environ 44.8%.

Dans la section 5, nous effectuerons des simulations de protocoles de diffusion de rumeur en temps discret. Nous observerons dans cette section qu'effectivement, de manière expérimentale, nous tendons vers des simulations où nous avons bien 44.8% des simulations qui présentent un temps de convergence supérieur à l'espérance.

4.4 Cas en temps continu

La structure de cette section est semblable à celle utilisée dans le cas discret. Dans un premier temps, nous écrivons une modélisation du problème. Dans un second temps, nous utilisons à nouveau cette modélisation afin d'analyser le comportement de la rumeur. Cette analyse, similaire à celle du cas en temps discret, est principalement axée sur le calcul de la probabilité d'effectuer une certaine interaction, et sur la réécriture du temps de convergence. Dans un troisième temps, nous nous attardons un peu sur les résultats présentés par Yves Mocquard concernant l'analyse du temps de convergence.

4.4.1 Réseau et états des différents nœuds

La modélisation est à nouveau effectuée à l'aide d'un graphe complet. La valeur associée à chaque nœud correspond au fait que la rumeur soit connue ou non par le nœud. Par simplicité, si le nœud connait la rumeur, sa valeur associée est égale à 1. Dans le cas contraire, cette valeur est égale à 0.

A l'instar du cas en temps discret, nous associons à chaque nœud une valeur à l'instant $t = 0$. Ces valeurs peuvent être soit 0, soit 1. Le nombre de nœuds associé à une valeur égale à 1 doit être strictement positif. Cela traduit le fait qu'au moins un nœud doit connaître la rumeur, initialement.

A la différence du cas en temps discret, nous avons $t \in \mathbb{R}^+$. L'état d'un nœud i à l'instant t est écrit $C_t^{(i)}$ pour tout $t \in \mathbb{R}^+$ et $i \in \{1, \ldots, n\}$. Ainsi, à l'instant $t = 0$ au moins un $C_0^{(i)}$ est égal à 1.

La fonction traduisant le changement d'état de deux nœuds interagissant entre eux est la même que celle du cas en temps discret (4.1), à savoir

$$\begin{aligned} f := \{0,1\}\times\{0,1\} &\to \{0,1\} \times \{0,1\} \\ (x,y) &\to (max\{x,y\}, max\{x,y\}) \end{aligned} \tag{4.30}$$

où (x, y) est donc le couple formé par les valeurs des deux nœuds qui interagissent, leur nouvelle valeur associée étant déterminée par la fonction $f(x, y)$.

A chaque nœud est associé un processus de Poisson indépendant, ces processus ayant tous le même paramètre $\lambda > 0$. Comme vu dans la sous-section 3.2.3, ces processus déterminent le temps séparant deux interactions initiées par un nœud. Par exemple, lorsque le processus de Poisson associé au nœud i a un saut, le nœud doit initier une interaction. Le nœud i choisit alors aléatoirement un autre nœud j pour interagir avec lui.

Le temps de convergence demeure le temps nécessaire pour diffuser la rumeur. Comme pour le cas en temps discret, il s'agit du temps qu'il faut afin que $C_t^{(i)}$ soit égal à 1 pour chaque $i \in \{1, \ldots, n\}$. La notation du temps de convergence dans le cas du protocole en temps continu est la suivante : Θ_n.

4.4.2 Protocole de diffusion de rumeur

Une fois la valeur de tous les nœuds initialisée, les interactions commencent à prendre place, rythmées par les différents processus de Poisson indépendant. Ces processus sont chacun liés au même taux λ. Ainsi, pour chaque nœud, le taux d'interactions initiées est défini par λ.

Comme nous travaillons avec n processus de Poisson de même paramètre λ, nous pouvons employer un processus de Poisson unique pour rythmer les interactions de tous les noeuds. Ce processus de Poisson est associé au taux $n\lambda$. Cela signifie que les sauts prennent place à un taux $n\lambda$, et qu'à chaque saut, deux nœuds distincts peuvent être choisis aléatoirement pour interagir. Le comportement de l'interaction en question est identique à celui pour le cas en temps discret.

Notons $(\tau_l)_{l\in\mathbb{N}} \in \mathbb{R}^+$ les instants auxquels les sauts du processus de Poisson de taux $n\lambda$ prennent place, où chacun de ces sauts est indexé par $l \in \mathbb{N}$. Considérons la l-ème interaction, entre les nœuds i et j. D'une manière similaire aux expressions (4.4) et (4.5), l'état des nœuds à l'issue de cette interaction est exprimé ainsi :

$$C_t^{(i)} = C_t^{(j)} = \max\{C_{\tau_{l-1}}^{(i)}, C_{\tau_{l-1}}^{(i)}\} \qquad t \in [\tau_l, \tau_{l+1}[\tag{4.31}$$

$$C_t^{(r)} = C_{\tau_{l-1}}^{(r)}, \qquad r \neq i, j, \;\; t \in [\tau_l, \tau_{l+1}[\tag{4.32}$$

Notons à présent X_l la variable aléatoire qui représente le choix du couple de nœuds (i, j) interagissant entre eux. Cette variable a un comportement identique à celui de X_t introduit à la section précédente dans l'expression (4.3). Dès lors, ce choix étant uniforme, nous décrivons X_l de la manière suivante :

$$\mathbb{P}\{X_l = (i, j)\} = \frac{1}{n(n-1)} 1_{i \neq j}, \qquad \forall i, j \in [\![1, n]\!] \tag{4.33}$$

Désormais, nous sommes en mesure de décrire, à partir d'un seul processus de Poisson, le taux d'interactions observées, la probabilité d'observer un certain choix parmi les nœuds lors d'une interaction, et l'issue de l'interaction en question.

Yves Mocquard note dans sa thèse Θ_n la variable aléatoire correspondant au temps de convergence dans le cadre du cas en temps continu. Cette variable aléatoire est définie comme ceci :

$$\Theta_n := \inf \left(t \geq 0 \middle| \sum_{i=1}^{n} C_t^{(i)} = n \right) \tag{4.34}$$

soit le temps nécessaire pour que la valeur associée à chaque nœud soit égale à 1.

Une représentation "simpliste" de l'évolution d'un réseau par un protocole de diffusion de rumeur en temps continu peut être effectuée comme le propose la figure 4.3. Initialement, un seul nœud connait la rumeur. En pointillés est représentée l'interaction prenant place à l'instant t. Sur le deuxième graphe, l'instant $t = \tau_1$ correspond à l'instant du premier saut du processus de Poisson de taux $n\lambda$ en $t = \tau_l$ avec $l = 1$.

4.4.3 Approche Markovienne

Regardons à présent dans quelle mesure nous pouvons traduire le modèle à l'aide d'une chaîne de Markov. A l'instar du cas discret, nous allons employer un processus stochastique. La différence est qu'il est ici à temps continu. Voici la définition d'un processus stochastique à temps continu :

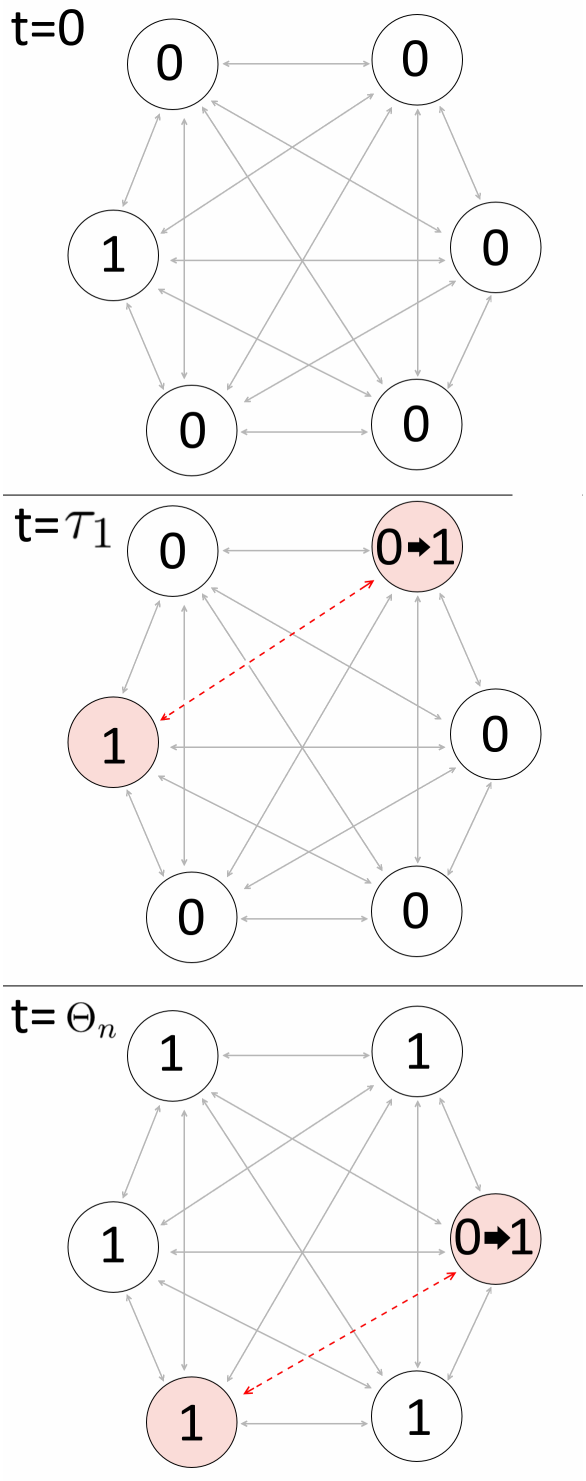

FIGURE 4.3 – Évolution d'un réseau de taille $n = 6$, en temps continu.

Définition - Processus stochastique à temps continu

Un processus stochastique à temps continu est une famille $X = \{X_t, t \in \mathbb{R}^+\}$ de variables aléatoires.

Introduisons le processus stochastique en temps continu $Z = \{Z_t, t \in \mathbb{R}^+\}$ à n états. Tout comme le processus stochastique Y introduit en (4.8), les états Z_t sont définis comme le nombre de nœuds connaissant la rumeur à un instant t. Autrement dit, cet état Z_t est défini comme :

$$Z_t := \sum_{i=1}^{n} C_t^{(i)} \tag{4.35}$$

Une chaîne de Markov en temps continu est un processus stochastique en temps continu pour lequel, pour chaque état, le processus change d'état après un temps issu d'une variable aléatoire exponentielle. Le changement donne lieu alors à un état différent, déterminé en fonction d'une matrice appelée générateur infinitésimal.

Ainsi, une chaîne de Markov en temps continu $X = \{X_t, t \in \mathbb{R}^+\}$ est définie par :
- un espace d'états dénombrable,
- une matrice de taux de transitions Q, aussi appelée générateur infinitésimal,
- un état initial k tel que $X_0 = k$.

Pour chaque élément $i \neq j$, l'élément Q_{ij} du générateur Q correspond aux taux de transitions de l'état i à l'état j. Ces taux sont positifs. Pour chaque état i, l'élément Q_{ii} est choisi de manière à respecter la contrainte suivante :

$$Q_{ii} = -\sum_{j \neq i} Q_{ij} \tag{4.36}$$

Pour chaque ligne de la matrice Q, la somme des éléments de cette ligne est donc égale à zéro. Pour chaque état i, l'élément $-Q_{ii}$ donne le paramètre de l'exponentielle déterminant l'instant associé au changement d'état depuis l'état i. L'interprétation du générateur infinitésimal est abordée de manière détaillée dans la section 7.1.2.

Le processus $Z = \{Z_t, t \in \mathbb{R}^+\}$ est une chaîne de Markov qui est caractérisée par une matrice de taux de transitions notée B, où les états sont définis sur $\{1, \ldots, n\}$ et où $Z_0 = 1$. Déterminons à présent les valeurs des éléments du générateur infinitésimal B. Cette matrice B exprime les taux de transitions d'un état à un autre.

Deux transitions sont possibles à l'issue d'une interaction, dans le cadre de la diffusion de rumeur. La première consiste à passer d'un état où i nœuds connaissent la rumeur à un état où $i+1$ nœuds connaissent la rumeur. La seconde transition est celle où on demeure dans l'état où i nœuds connaissent la rumeur à partir du même état, c'est-à-dire $Z_{\tau_l} = Z_{\tau_{l+1}}$. Les taux associés à chacune de ces deux transitions sont exprimés de la manière suivante :

$$\begin{aligned} B_{i,i} &= -n\lambda p_i \\ B_{i,i+1} &= n\lambda p_i, \qquad i \neq n \end{aligned} \tag{4.37}$$

où $i \in \{1, \ldots, n\}$, et où les autres valeurs de la matrice sont nulles. La formule de p_i est définie dans l'expression 4.16, et trouve la même justification que celle présentée pour l'expression 4.16. Lorsque $Z_t = i$, on a que le prochain nœud initiant une interaction est sollicité avec l'intensité $n\lambda$.

Finalement, sous cette modélisation, on exprime le temps de convergence Θ_n comme

$$\Theta_n = \inf\{t \geq 0 | Z_t = n\} \tag{4.38}$$

Cette formule clôture cette sous-section, nous permettant à présent, à partir d'un état initial Z_0, de faire évoluer la chaîne de Markov $Z = \{Z_t, t \in \mathbb{R}^+\}$. Cela est effectué par le biais des taux de transitions définis dans la matrice B, nous donnant l'intensité associée aux différentes transitions, jusqu'à ce que l'état de convergence soit atteint en Θ_n.

Chapitre 5

Simulation du protocole de diffusion de rumeur

Ce chapitre propose une implémentation d'une simulation d'un protocole de diffusion de rumeur lorsque le temps est discrétisé. Dans un premier temps, une présentation du modèle est effectuée. Nous proposons, dans une analyse, des résultats obtenus suite à l'implémentation du modèle.

5.1 Présentation du modèle

Cette simulation concerne le cas du modèle discret décrit dans la section 4.2. Pour rappel, nous avons une population d'individus, dont certains connaissent initialement une rumeur. Les individus interagissent deux par deux. Lors d'une interaction mettant en scène une personne qui connait la rumeur et une qui ne la connait pas, la rumeur se propage d'un individu à l'autre. Les interactions prennent place tour à tour. La paire d'individus concernée par une interaction est choisie aléatoirement et uniformément parmi l'ensemble des individus.

5.1.1 Objectifs

L'objectif principal est d'estimer l'espérance du temps de convergence T_n, à savoir le nombre d'itérations nécessaires pour que l'ensemble de la population connaisse la rumeur, sur base de la taille de la population et du nombre d'individus connaissant initialement la rumeur.

5.1.2 Modèle conceptuel

Sur base des objectifs décrits ci-dessus nous présentons à présent la structure de données adoptée pour réaliser la simulation.

Une approche naïve peut être de considérer chaque individu individuellement, comme étant une entité possédant deux états :

— l'individu connait la rumeur
— l'individu ne connait pas la rumeur.

L'outil de simulation pourrait alors ensuite choisir aléatoirement deux entités parmi l'ensemble des entités afin de les faire interagir.

Cette approche présente cependant un problème : lors d'une taille de la population trop élevée, le temps d'exécution peut devenir très long de par une utilisation inadéquate des accès de mémoire. C'est pourquoi nous adoptons une structure plus adaptée.

L'état du système à un certain instant t peut être résumé en deux informations. La première est l'instant t qui correspond au fait que le système soit à la t-ième interaction depuis le début de la simulation. La seconde variable est le nombre d'individus connaissant la rumeur, soit Y_t (introduit avec l'expression (4.8)). Comme présenté dans la section 4.2.3, ce nombre évolue au fil des interactions, avec une probabilité p_i de se voir augmenter de 1. L'expression de cette probabilité p_i est reprise dans la sous-section ci-dessous.

Le système étant fondamentalement élémentaire, nous n'avons que deux paramètres agissant comme conditions initiales : la taille de la population n et le nombre d'individus connaissant initialement la rumeur Y_0.

5.1.3 Modèle de spécifications

Nous présentons à présent les spécifications qui viennent régir l'évolution du système, sur base du modèle conceptuel que nous venons d'introduire. Nous souhaitons donc décrire le comportement des variables principales ainsi que leurs éventuelles interactions.

L'instant t, qui compte également les interactions, est un naturel dont la valeur initiale est 0. Cette horloge est incrémentée à chaque instant, et donc à chaque interaction effectuée.

Notre variable d'état Y_t, donnant le nombre d'individus connaissant la rumeur à un instant t, est un naturel compris entre 1 et n, n étant la taille de notre population :

$$Y_t = \sum_{i=1}^{n} C_t^{(i)} \tag{5.1}$$

où $C_t^{(i)}$ est l'état associé au i-ème individu à l'instant t (1 si l'individu connait la rumeur, et 0 sinon).

A chaque interaction successive, la variable Y_t peut soit être augmentée de 1, soit conserver sa valeur. L'issue de l'interaction est associée à la probabilité p_i :

$$p_i := \mathbb{P}\{Y_{t+1} = i+1 | Y_t = i\} = \frac{2i(n-i)}{n(n-1)} \tag{5.2}$$

Sur la figure 5.1 est représenté un graphe de transition pour décrire l'évolution du processus $\{Y_t, t \in \mathbb{N}\}$ au fur et à mesure des interactions en fonction des probabilités p_i.

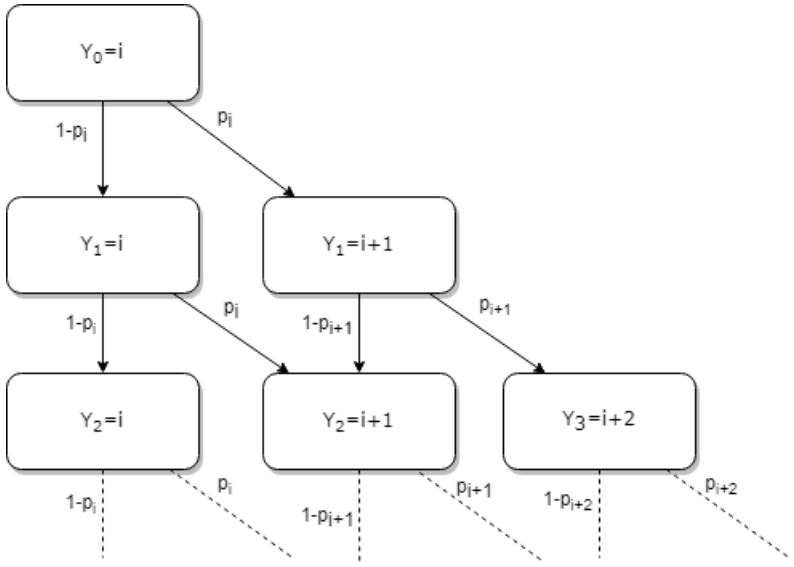

FIGURE 5.1 – Évolution du processus $\{Y_t, t \in \mathbb{N}\}$ au fil du temps

Pour le bon déroulement de la simulation, il est nécessaire que $n \in \mathbb{N}_0$, et que $Y_0 \in \{1, \ldots, n-1\}$.

5.1.4 Implémentation

L'implémentation est ici réalisée à l'aide du langage Python. Ce choix a été motivé par le fait que Python offre des librairies particulièrement adaptées à l'implémentation de simulations d'évènements discrets (par exemple NumPy et SciPy). Cependant, il n'est pas exclu d'utiliser des langages proposant des outils équivalents, comme le fait par exemple R ou matlab.

Afin d'obtenir le temps de convergence T_n lors d'une simulation, il est nécessaire d'enchaîner les interactions jusqu'à ce que l'état de convergence soit atteint. Autrement dit, après avoir initialisé Y_0 à 1 (comme discuté dans la sous-section 5.1.6), une boucle prend place, où l'itération consiste à incrémenter Y_t avec une certaine probabilité p_i où $i = Y_t$. La condition d'arrêt est alors $Y_t = n$, traduisant un état de convergence atteint. A chaque itération, nous prenons soin d'incrémenter t qui représente le temps écoulé.

Pour le calcul du temps de convergence, il nous suffit de conserver la variable t une fois l'état de convergence atteint.

5.1.5 Validation

Une fois notre modèle implémenté, il est nécessaire de vérifier que son comportement soit bien cohérent avec celui souhaité. Si la taille de la population augmente, nous souhaitons voir augmenter le temps de convergence moyen, et inversement.

Dans le chapitre précédent, nous avons établi une expression pour la distribution de T_n en (4.25). Pour rappel, l'espérance de T_n pouvait être formulée ainsi :

$$\mathbb{E}(T_n|Y_0 = i) = \frac{(n-1)(H_{n-1} + H_{n-i} - H_{i-1})}{2} \tag{5.3}$$

où H_k est la k-ième somme partielle de la série harmonique.

Nous sommes dès lors en mesure de comparer les espérances estimées à l'aide des simulations et celles souhaitées à l'aide d'un test d'hypothèse sur l'égalité de l'espérance avec une valeur fixée.

De plus, nous pouvons utiliser les propriétés établies sur les bornes de la distribution de T_n afin de vérifier que lorsque $Y_0 = 1$, l'expression (4.29) est respectée. Autrement dit, nous devons observer la convergence suivante :

$$\lim_{n\to\infty} \mathbb{P}\{T_n > \mathbb{E}(T_n)\} = 0.448429663727 \tag{5.4}$$

5.1.6 Exécution de la simulation

Le paramètre Y_0 est initialisé à 1 dans le cadre de nos simulations, afin de pouvoir éventuellement tester les propriétés liées aux bornes de la distribution de T_n, comme discuté dans la section 4.3.

Afin de pouvoir étudier les résultats acquis à l'aide des simulations, nous sommes amenés à devoir former un échantillon $\underline{X}$ du temps de convergence composé de nombreuses simulations indépendantes. Sur base d'une observation $\underline{x}$, nous pouvons obtenir un estimateur de l'espérance de la variable T_n. Le nombre de simulations effectuées pour l'estimation du temps de convergence est de 10^5. Une fois l'estimation de $\mathbb{E}(T_n)$ acquise, nous sommes libres d'effectuer différents tests statistiques afin de valider notre outil de simulation.

Nous collectons donc le temps de convergence observé pour chacune des simulations dans une liste. L'analyse des résultats est effectuée en trois parties :

1. Nous regardons les valeurs obtenues pour les estimations de $\mathbb{E}(T_n)$, en prenant la moyenne des temps de convergence observés. Nous calculons également l'écart-type empirique corrigé afin d'avoir un premier indicateur sur la dispersion des valeurs de T_n par rapport à l'espérance. Finalement, nous utilisons le coefficient de variation afin de compléter notre analyse de la dispersion des valeurs par rapport à l'espérance.
2. Nous effectuons un test d'hypothèse sur l'espérance de T_n pour un n fixé afin de vérifier que l'estimateur de $\mathbb{E}(T_n)$ n'est pas différent de la valeur souhaitée.
3. Nous calculons la proportion des temps de convergence T_n supérieurs à $\mathbb{E}(T_n)$ parmi les simulations, et regardons si elle converge vers ≈ 0.4484 à mesure que $n \to \infty$. Pour ce faire, nous regardons l'évolution des intervalles de confiance à 95% sur cette même proportion.

A l'issue de chacune de ces parties, si les valeurs calculées ne sont pas jugées acceptables, nous pouvons augmenter le nombre de simulations effectuées afin d'espérer obtenir des résultats plus adéquats. Dans l'analyse de la convergence, nous pouvons également augmenter la valeur maximum du n qui a un impact sur le temps de convergence puisque plus d'interactions seront nécessaires. Si en revanche les valeurs sont jugées satisfaisantes, nous considérerons la simulation comme validée.

5.2 Résultats

Sur base du modèle que nous avons défini, observons à présent les résultats que nous obtenons au cours des différentes simulations. D'abord, nous tentons d'estimer l'espérance de T_n, en vérifiant si les simulations obtenues sont cohérentes avec les résultats théoriques établis dans le chapitre précédent. Ensuite, nous effectuons un test d'hypothèse sur l'espérance de T_n pour un n fixé afin de se convaincre que l'espérance n'est pas différente de celle théorique. Finalement, nous voulons observer la convergence de $\mathbb{P}\{T_n > \mathbb{E}(T_n)\}$ vers 0.448429663727. Par simplicité, Y_0 a été fixé à 1 lors de la réalisation des simulations.

5.2.1 Estimation de l'espérance de T_n

Sur la figure 5.2, nous représentons en pointillés bleus les valeurs estimées de $\mathbb{E}(T_n)$ sur base des simulations pour différentes tailles de population. En orange est représentée la courbe associée aux valeurs théoriques de $\mathbb{E}(T_n)$. La zone en bleu représente les intervalles de confiance à 95% sur l'espérance, calculés comme ceci pour chaque échantillon $\underline{x}$ réalisé :

$$\left[\hat{\mu}_n(\underline{x}) - t_{n-1;\frac{1.95}{2}} \frac{\hat{S}_{n,c}(\underline{x})}{\sqrt{n}}, \hat{\mu}_n(\underline{x}) - t_{n-1;\frac{0.05}{2}} \frac{\hat{S}_{n,c}(\underline{x})}{\sqrt{n}}\right] \tag{5.5}$$

où $\hat{\mu}_n(\underline{x})$ est l'estimation de $\mathbb{E}(T_n)$, $\hat{S}_{n,c}(\underline{x})$ est l'écart-type empirique corrigé et $t_{n,\alpha}$ est le quantile d'ordre α de la loi Student à n degrés de liberté.

Nous remarquons sur la figure 5.2 qu'à mesure que la taille de la population augmente, le nombre d'interactions moyen requis pour atteindre l'état de convergence augmente également, comme souhaité. Nous remarquons également que l'estimation semble à première vue être une bonne approximation de la valeur souhaitée, même s'il nous faut davantage d'arguments pour s'en convaincre. Finalement, nous notons que l'intervalle de confiance semble augmenter à mesure que la taille de la population augmente. Cela peut être traduit par des valeurs de plus en plus éloignées de l'espérance.

Afin d'observer plus clairement la différence entre la courbe de l'estimateur et de la valeur théorique, nous représentons dans la figure 5.3 la valeur absolue de l'erreur entre les deux courbes, à l'aide d'une échelle logarithmique sur l'axe y.

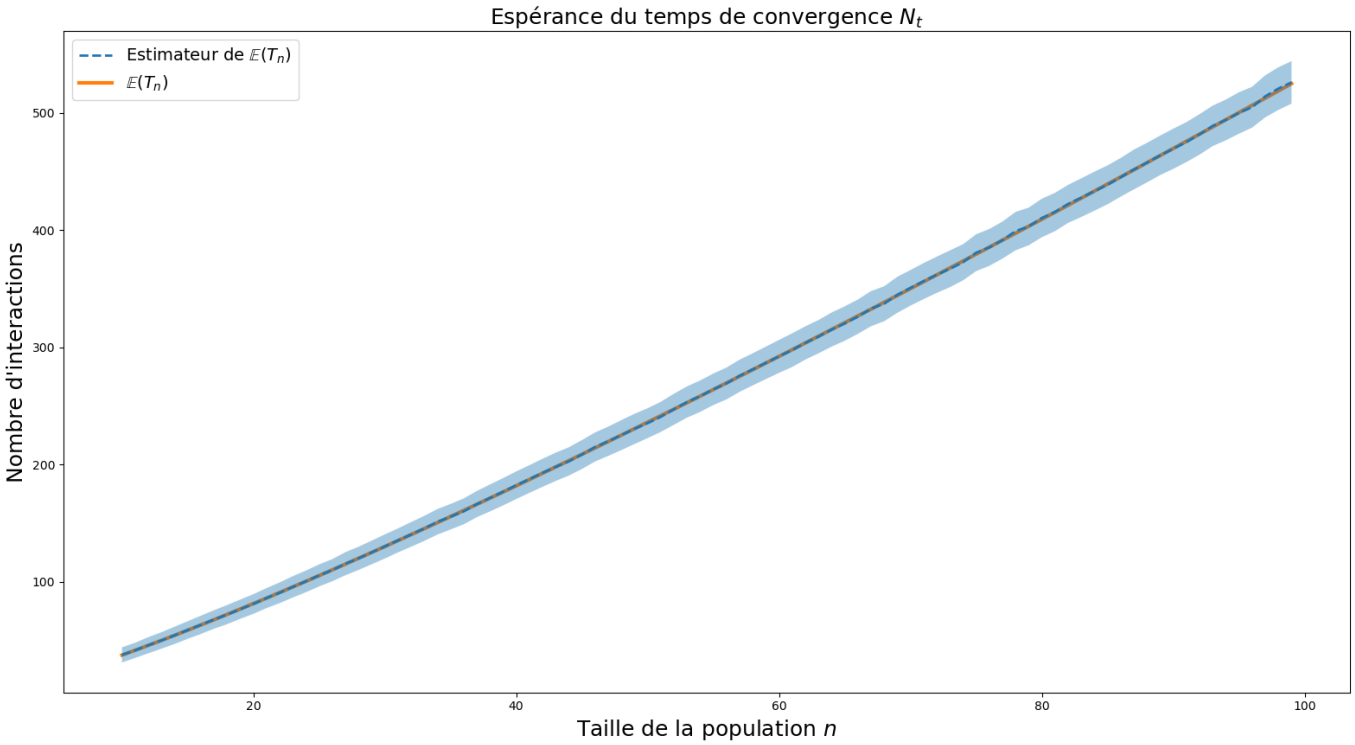

FIGURE 5.2 – Comparaison de $\mathbb{E}(T_n)$ théorique et de son estimation

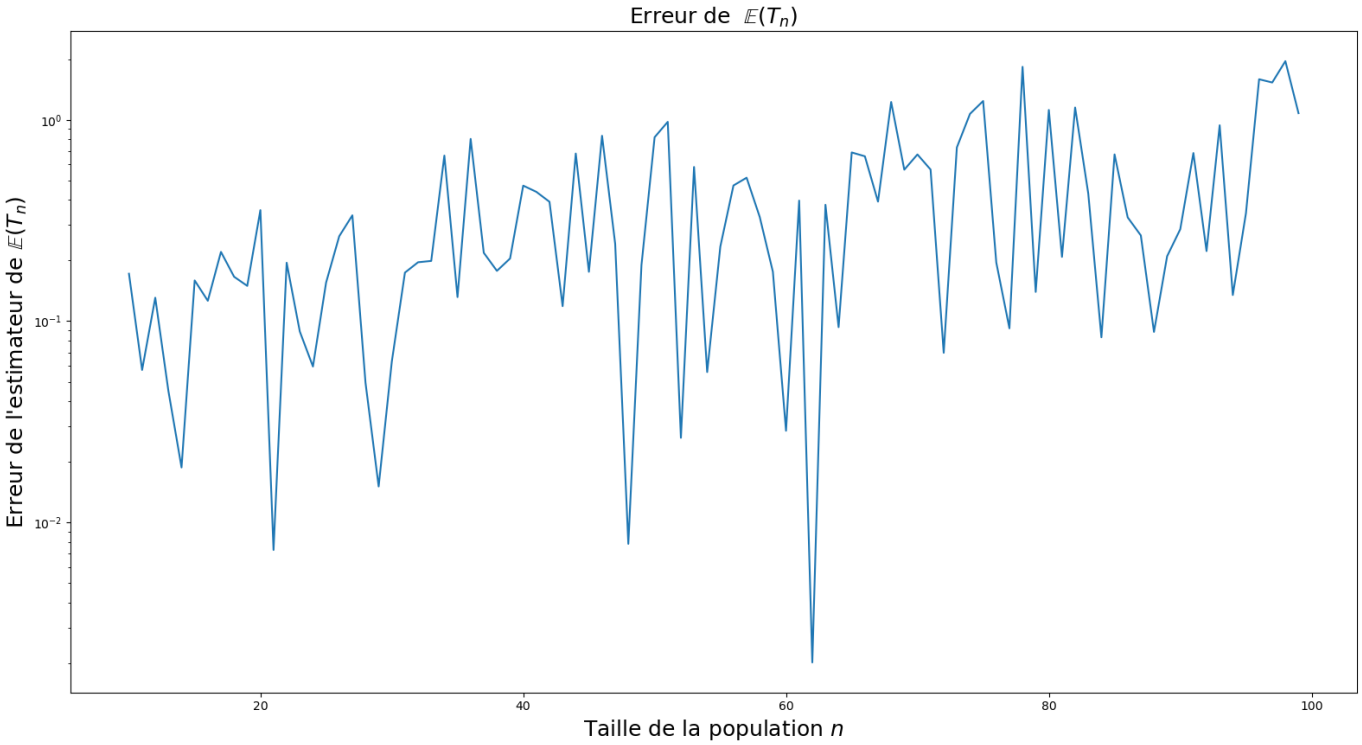

FIGURE 5.3 – Erreur de $\mathbb{E}(T_n)$

5.2.2 Test sur l'espérance de T_n

Fixons n à 100 et considérons l'échantillon $\underline{X}$ des temps de convergence obtenus lors de l'exécution de 10^5 simulations. La valeur théorique peut aisément être obtenue à l'aide de l'expression 4.25. Comme nous sommes dans un cas où $Y_0 = 1$, cette expression se simplifie en :

$$\mathbb{E}(T_n) = (n-1)H_{n-1} \tag{5.6}$$

où pour rappel, H_k est la k-ième somme partielle de la série harmonique. Dans le cas où $n = 100$, $H_{n-1} \approx 5.177$. Dès lors, la valeur théorique de $\mathbb{E}(T_n)$ est de 512.56.

Soit notre observation $\underline{x}$ de taille 10^5, testons si l'estimation de l'espérance de T_n est différente de 512.56 avec 5% de chance de me tromper. Nous faisons donc appel à un test bilatéral de l'espérance d'une population quelconque de variance inconnue. Nos hypothèses seront les suivantes :

— $\mathcal{H}_0 : \mu = \mathbb{E}(T_n)$,
— $\mathcal{H}_1 : \mu \neq \mathbb{E}(T_n)$.

La statistique de test est la suivante :

$$T(\underline{x}) = \frac{\hat{\mu}_n - \mu_0}{\hat{S}_{n,c}(\underline{x})/\sqrt{n-1}} \tag{5.7}$$

où $\hat{\mu}_n$ est la moyenne arithmétique calculée sur base de $\underline{x}$, et $\hat{S}_{n,c}(\underline{x})$ est l'écart-type empirique corrigé sur base de $\underline{x}$.

Regardons si

$$T(\underline{x}) \notin [-z_{1-\alpha/2}, z_{1-\alpha/2}] \tag{5.8}$$

avec $\alpha = 0.05$ et z_u est le quantile d'ordre u pour une distribution normale centrée réduite. Si c'est le cas, nous pourrons rejeter notre hypothèse $\mathcal{H}_0$.

Ici, $T(\underline{x}) \approx 0.169$ et $[-z_{1-\alpha/2}, z_{1-\alpha/2}] \approx [-1.96, 1.96]$. Dès lors, notre statistique de test se situant dans l'intervalle de confiance, nous ne pouvons pas rejeter notre hypothèse nulle avec 5% de chance de se tromper. Même s'il s'agit d'un indicateur nous poussant à croire que l'estimation de T_n n'est pas radicalement éloignée de sa valeur théorique, il est cependant très important de préciser que cela ne veut pas nécessairement dire que H_0 est à accepter sans risque. En effet, il faut considérer l'erreur de seconde espèce β si nous souhaitons accepter H_0 avec β chances de se tromper.

5.2.3 Convergence de $\mathbb{P}\{T_n > \mathbb{E}(T_n)\}$

Dans le chapitre précédent, nous avons vu avec l'expression 4.29 que la probabilité d'observer un temps de convergence supérieur à son espérance convergeait vers approximativement 0.448, à la condition que $Y_0 = 1$. Tentons d'observer cette convergence à l'aide de nos simulations.

Dans un premier temps, pour chaque n situé entre 15 et 85, nous exécutons 10^5 simulations indépendantes de manière à avoir 70 estimations de l'espérance du temps de convergence. Ensuite, nous calculons la proportion de simulations où le temps de convergence est supérieur à l'espérance théorique. Nous calculons finalement pour chaque n l'intervalle de confiance à 95% de la proportion de ces temps.

Sur la figure 5.4, la courbe bleue représente les proportions introduites ci-dessus, à savoir l'estimation de la proportion de $T_n > \mathbb{E}(T_n)$ sur base des simulations $\underline{x}$. La ligne horizontale rouge représente la valeur vers laquelle on souhaite converger. Finalement, la zone en bleu enveloppant la courbe bleue représente les intervalles de confiance liés à la courbe bleue. Ces intervalles de confiance sont calculés pour chaque échantillon $\underline{x}$

comme ceci :

$$\left[\hat{\pi}_n(\underline{x}) + z_{\frac{0.05}{2}}\sqrt{\frac{\hat{\pi}_n(\underline{x})(1-\hat{\pi}_n(\underline{x}))}{n}}, \hat{\pi}_n(\underline{x}) + z_{\frac{1.95}{2}}\sqrt{\frac{\hat{\pi}_n(\underline{x})(1-\hat{\pi}_n(\underline{x}))}{n}}\right] \tag{5.9}$$

où $\hat{\pi}_n(\underline{x})$ est l'estimateur de la proportion de simulations où le temps T_n est supérieur à l'espérance théorique, et z_α est le quantile d'ordre α de la loi normale centrée réduite.

On remarque sur la figure 5.4 que l'estimation de $\mathbb{P}\{T_n > \mathbb{E}(T_n)\}$ oscille autour de la valeur souhaitée. Cependant, nous pouvions espérer un rétrécissement de l'intervalle de confiance à mesure que n augmente, resserrant la courbe autour de 0.448, mais nous ne voyons pas de phénomène semblable sur le graphe.

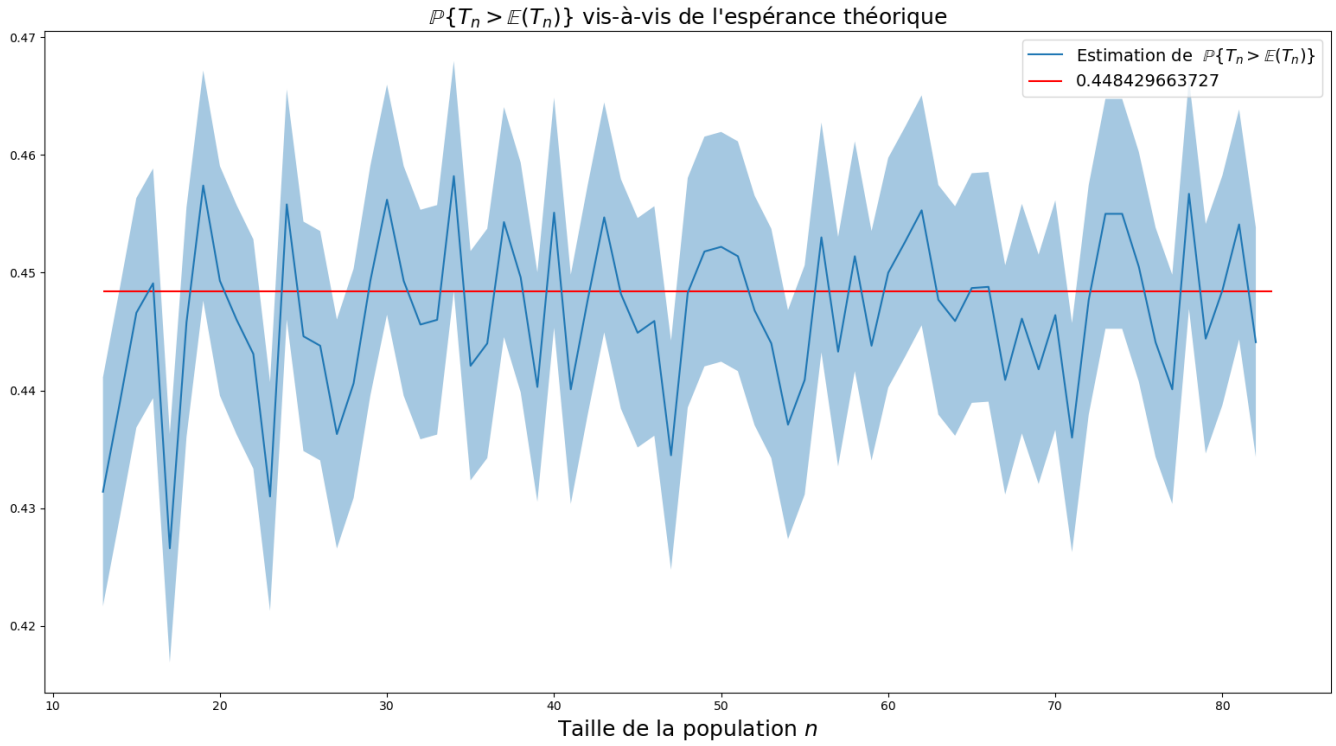

FIGURE 5.4 – Probabilité d'observer un temps de convergence supérieur à son espérance théorique.

Nous souhaitons cependant voir si une convergence plus marquée est présente si nous augmentons le nombre de simulations. Sur la figure 5.5, nous représentons la même chose que sur la figure 5.4 à la différence que le nombre de simulations effectuées pour chaque estimation est de 10^6.

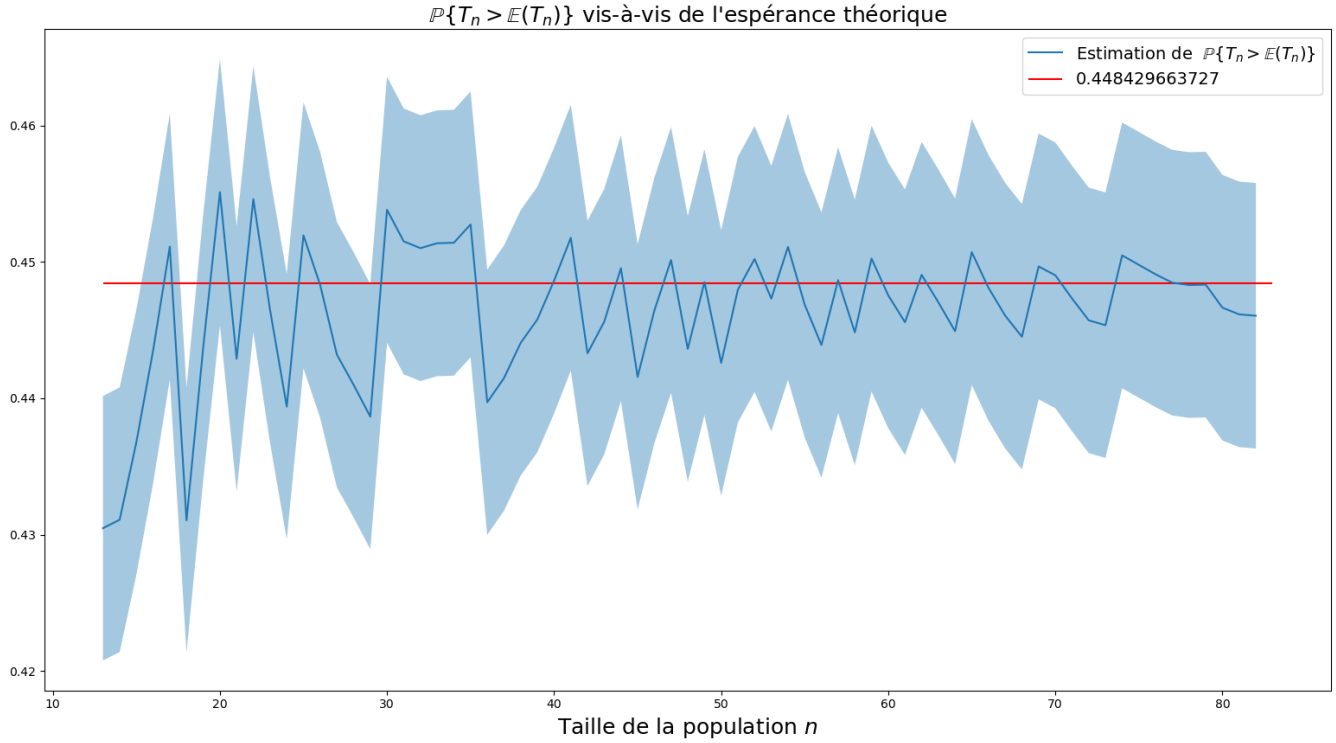

FIGURE 5.5 – Probabilité d'observer un temps de convergence supérieur à son espérance théorique.

Il semble qu'augmenter le nombre de simulations d'un facteur 10 nous permette d'observer une tendance plus proche de celle souhaitée. L'intervalle de confiance semble également moins osciller autour de la ligne horizontale rouge. Afin d'observer de manière plus claire cette tendance, nous comparons les figures 5.4 et 5.5 à l'aide d'un graphe d'erreur. Ce graphe est visible sur la figure 5.6, où nous représentons la valeur absolue des erreurs entre les courbes visibles sur les figures 5.4 et 5.5 à l'aide d'une échelle logarithmique sur l'axe y. La courbe bleue correspond au cas où on effectue 10^5 simulations par estimation, et celle orange correspond au cas où on en effectue 10^6.

Nous pouvons remarquer qu'effectivement, la courbe orange présente une légère tendance à décroître, là où la courbe bleue ne semble pas évoluer.

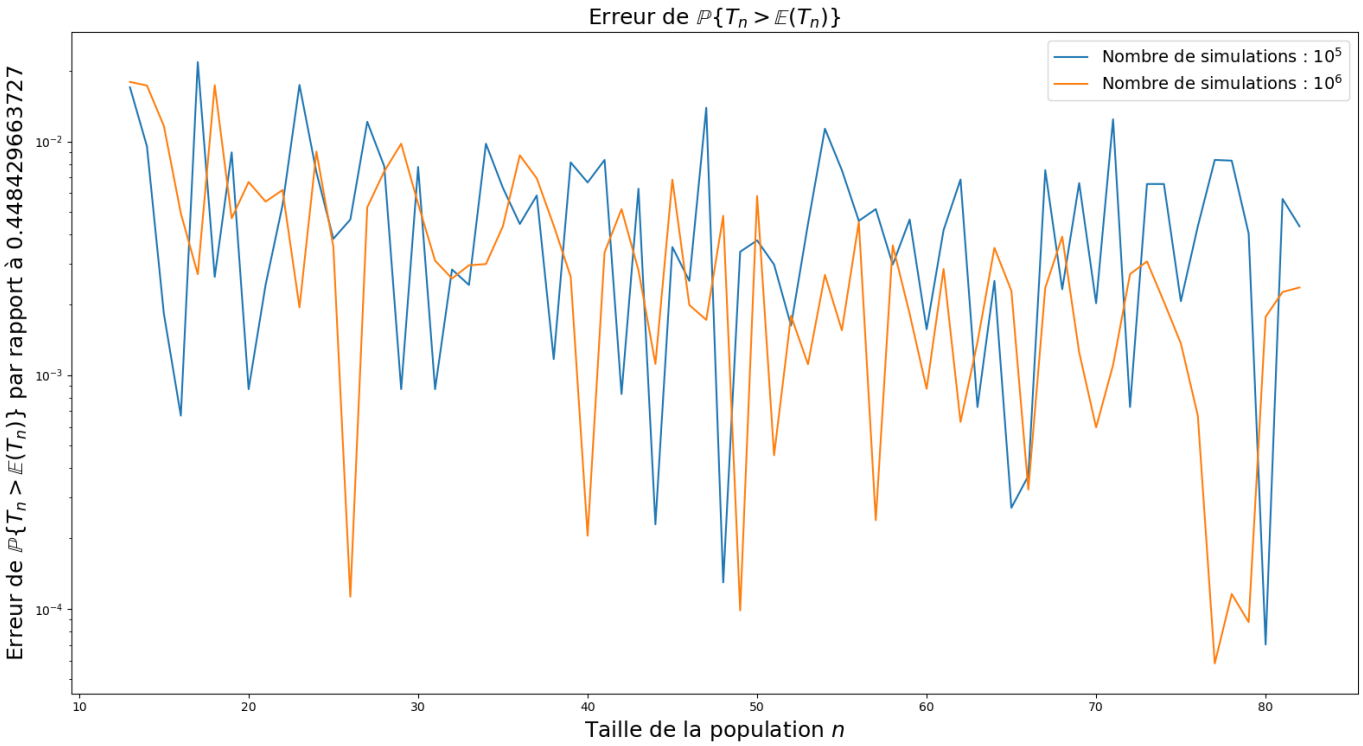

FIGURE 5.6 – Erreur sur la proportion d'estimation des temps de convergence supérieurs à leur espérance théorique.

Chapitre 6

Markovian Arrival Process

6.1 Introduction

Ce chapitre porte sur le "Markovian Arrival Process", appelé MAP. Afin de pouvoir aborder ce processus, plusieurs notions sont abordées en amont. Nous commençons par décrire ce qu'est un processus de renouvellement, en l'introduisant avec le processus de Poisson. Ensuite, nous abordons la description d'une distribution de type phase. Ces deux notions nous permettent de décrire le processus de renouvellement de type phase. Finalement, nous décrivons le Markovian Arrival Process. L'étude effectuée dans ce chapitre s'appuie principalement sur l'ouvrage [10] de G. Latouche et V. Ramaswami.

6.2 Processus de renouvellement

Un processus de renouvellement peut être perçu comme une généralisation du processus de Poisson.

6.2.1 Processus de Poisson

Un processus de Poisson est une chaîne de Markov $X = \{X_i, i \in \mathbb{N}\}$ à temps continu. Pour rappel, la notion de chaîne de Markov est introduite au chapitre 4 dans la section 4.4.3. Le processus est donc une collection de variables aléatoires notées ainsi :

$$\{X_i, i \in \mathbb{N}_0\} \tag{6.1}$$

où X_i correspond au temps entre le i-ème événement et le précédent et où X_1 correspond au temps séparant le début et le premier événement. Dans un tel processus, les

temps X_i entre deux événements sont des variables aléatoires de loi commune exponentielle de paramètre λ, où λ est une constante. Ces variables X_i sont indépendantes les unes des autres.

Le processus de Poisson peut être interprété comme un processus de comptage. Pour rappel, un processus de comptage est un processus stochastique à valeurs dans $\mathbb{N}$ modélisant un nombre entier évoluant à travers le temps. Il est alors représenté à l'aide de la notation suivante :

$$\{N_t, t \geq 0\} \tag{6.2}$$

La valeur de N_t augmente avec chaque événement qui prend place. Cette valeur compte dès lors le nombre d'événements ayant eu lieu depuis l'instant 0 jusque t. Les temps entre deux événements sont indépendants des autres événements. Ces derniers sont définis par la collection de variables aléatoires $\{X_i, i \in \mathbb{N}_0\}$ définis en (6.1). Lorsque t est égal à 0, nous avons $N_0 = 0$. La figure 6.1 représente une évolution possible de N_t au fil du temps.

Comme les temps inter-événements sont associés à des variables aléatoires exponentielles de paramètre λ, le nombre d'événements prenant place sur un intervalle de temps t est une variable aléatoire de Poisson de paramètre λt.

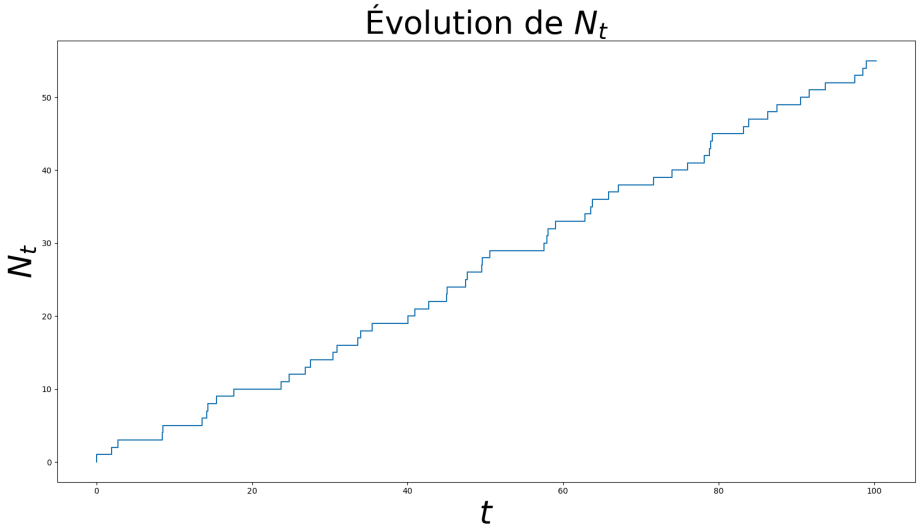

FIGURE 6.1 – Évolution de N_t

Étant donné que autant $\{X_i, i \in \mathbb{N}_0\}$ que $\{N_t, t \geq 0\}$ peuvent caractériser le processus de Poisson, nous pouvons trouver des liaisons entre les deux approches. La figure 6.2

présente un exemple d'évolution d'un processus de Poisson. Dans cette figure, nous observons l'incrémentation de N_t au fur et à mesure des événements. Nous observons également que le temps séparant chaque événement est défini par X_i.

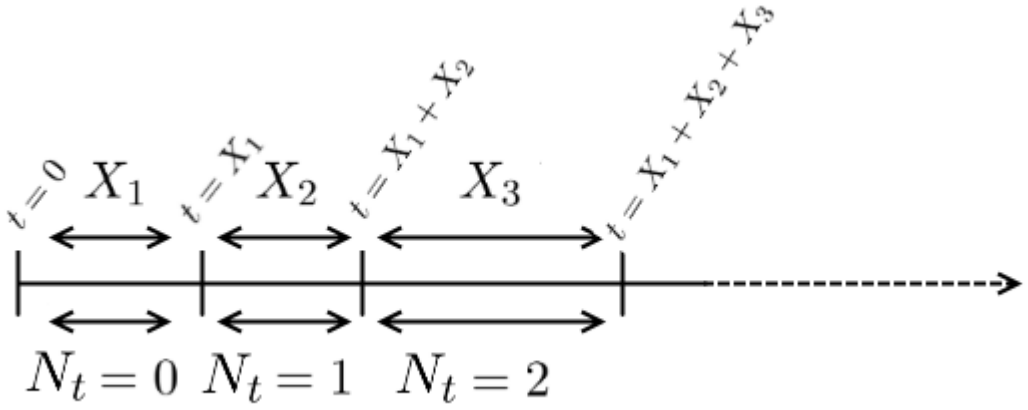

FIGURE 6.2 – Exemple d'évolution d'un processus de Poisson

6.2.2 Définition du processus de renouvellement

Maintenant que le processus de Poisson est défini, regardons comment le généraliser de manière à définir un processus de renouvellement.

Désormais, l'idée n'est plus d'utiliser une loi exponentielle pour définir les variables X_i, mais d'autoriser l'emploi d'une autre loi. Un processus de renouvellement est donc également un processus de comptage, dont la différence avec le processus de Poisson est la définition du temps inter-événements.

De manière plus formelle, un processus de comptage dont la suite des temps inter-événements forme une suite de variables aléatoires indépendantes et identiquement distribuées s'appelle un processus de renouvellement.

Il peut être caractérisé de différentes manières :

- la première manière est la collection de variables $X_i, i \in \mathbb{N}_0$ correspondant aux temps inter-événement
- une seconde manière est d'utiliser la notation $\{N_t, t \geq 0\}$ introduite en (6.2) correspondant au nombre d'événements ayant eu lieu jusqu'à l'instant t depuis le début (en $t = 0$)

— finalement, une dernière manière pourrait être d'employer une collection de variables aléatoires $T_i, i \in \mathbb{N}_0$ correspondant aux instants associés aux événements.

Dans le cadre des processus de renouvellement, ce que nous appelions jusqu'ici un événement est également appelé renouvellement.

6.3 Distribution de type phase

Le processus de renouvellement étant désormais défini, nous présentons à présent une approche de la notion de distribution de type phase.

Considérons un processus de Markov défini sur les états $\{0, 1, \ldots, n\}$, avec $n \in \mathbb{N}$ fini. A la différence des chaînes de Markov, un processus de Markov peut être à temps continu. Associons au processus de Markov précédemment considéré un générateur infinitésimal (aussi appelé matrice de taux de transitions) Q et un vecteur de probabilités initial noté (τ_0, τ).

L'interprétation que nous pouvons faire des éléments q_{ij} de la matrice Q, pour $i \neq j$, est qu'ils représentent la vitesse de transition de l'état i vers l'état j. Ce sont dès lors des réels positifs. Pour une distribution de type phase, la matrice Q possède la structure suivante :

$$Q = \begin{bmatrix} 0 & 0 \\ t & T \end{bmatrix} \tag{6.3}$$

où T est une matrice $n \times n$, et t est un vecteur colonne de taille n, choisi de manière telle que

$$T\mathbf{1} + t = 0 \tag{6.4}$$

avec $\mathbf{1}$ un vecteur colonne dont les éléments sont tous égaux à 1. Nous remarquons que l'état 0, associé à la première ligne de la matrice Q, est un état absorbant. Le vecteur t est donc associé à la vitesse des transitions vers l'état absorbant 0 depuis

chacun des autres états.

Le vecteur de probabilité initial (τ_0, τ) mentionné plus haut représente les probabilités de commencer dans un certain état. Il est donc logique d'avoir la contrainte suivante :

$$\tau_0 + \tau \mathbf{1} = 1 \tag{6.5}$$

L'élément τ_0 correspond à la probabilité de commencer dans l'état absorbant 0. Les éléments τ_i du vecteur τ correspondent respectivement aux probabilités de commencer dans les états $\{1, 2, \ldots, n\}$. Les éléments τ_i pour $0 \leq i \leq n$ sont donc tous des réels positifs.

Le temps X nécessaire pour atteindre l'état absorbant 0 est une variable aléatoire appelée variable aléatoire de type phase. La distribution de type phase est la distribution du temps X écoulé jusqu'à l'absorption. Cette distribution est notée $PH(\tau, T)$.

Example 6.3.1. *Considérons un ordinateur pouvant être sujet à plusieurs types d'erreur. Lorsque l'ordinateur est neuf, on considère que le temps nécessaire à attendre pour obtenir une erreur est distribué comme une variable aléatoire exponentielle dont la moyenne est de 100 heures. Trois types d'erreur sont envisageables. L'erreur de type 1 peut arriver avec une probabilité de 0.5. L'erreur de type 2 peut arriver avec une probabilité de 0.4. Finalement, l'erreur de type 3 peut arriver avec une probabilité de 0.1. L'erreur de type 1 fait retourner l'ordinateur dans son état d'origine. L'erreur de type 2 fait baisser le temps moyen avant la prochaine erreur de 50 heures et change les probabilités associées aux prochaines erreurs à 0.75, 0.2 et 0.05 respectivement. L'erreur de type 3 est fatale pour l'ordinateur.*

L'espérance de vie de l'ordinateur est définie comme une variable aléatoire de type phase. Outre l'état absorbant, nous définissons 4 états de la manière suivante :

$$T = \begin{bmatrix} -0.01 & 0.005 & 0.004 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0.015 & 0.004 & -0.02 \end{bmatrix} \tag{6.6}$$

Le premier état est l'état où on considère l'ordinateur comme neuf. Le second est associé à l'erreur de type 1, renvoyant immédiatement l'ordinateur à l'état 1. Le troisième état correspond à l'erreur de type 2, suite à laquelle l'ordinateur est dans l'état 4. L'état 4 correspond au fonctionnement de l'ordinateur à la suite d'une erreur de type 2.

Comme nous commençons avec un ordinateur neuf, nous avons

$$\tau = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.7}$$

Si nous devions représenter le générateur infinitésimal Q, celui-ci serait le suivant :

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.001 & -0.01 & 0.005 & 0.004 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0.001 & 0 & 0.015 & 0.004 & -0.02 \end{bmatrix} \tag{6.8}$$

et le vecteur de probabilités initial (τ_0, τ) serait

$$(\tau_0, \tau) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{6.9}$$

6.4 Processus de renouvellement de type phase

Nous avons à présent abordé les pré-requis pour introduire les processus de renouvellement de type phase.

Le processus de renouvellement de type phase est le processus

$$\{(N_t, J_t), t \in \mathbb{R}_0^+\} \tag{6.10}$$

où N_t donne le nombre de renouvellements ayant eu lieu dans l'intervalle de temps $(0, t)$ et J_t donne la phase dans laquelle se trouve le système à l'instant t. En tout temps $t \in \mathbb{R}_0^+$, nous avons que la phase $J_t \in \{1, 2, \ldots, n\}$. La phase correspond à ce que nous appelions état dans la section 6.3.

Nous avons donc un processus de Markov $\{J(t), t \geq 0\}$ associé au générateur Q défini en (6.3), où nous considérons X comme étant le temps écoulé jusqu'à l'absorption en phase 0. L'espace d'états de (N_t, J_t) est $\mathbb{N}_0 \times \{1, 2, \ldots, n\}$.

Depuis chaque état de l'espace d'états, voici les transitions possibles :

- les transitions avec un renouvellement, de l'état (m, i) à $(m + 1, j)$, le nombre de renouvellements maximum lors d'une transition étant de 1
- les transitions sans renouvellement, de l'état (m, i) à l'état (m, j).

Lorsque nous sommes sur une transition sans renouvellement, cela signifie que l'état absorbant n'a pas été atteint. La matrice T définie en (6.3) détermine vers quelle phase la transition est effectuée.

Si nous sommes dans la phase i, l'élément t_i du vecteur t défini en (6.3) détermine la vitesse à laquelle il y a une absorption, et donc un renouvellement. Une fois que le renouvellement a eu lieu, un nouvel intervalle débute, distribué comme une variable aléatoire de type phase. La nouvelle phase est alors décidée en fonction du vecteur de probabilités initial (τ_0, τ). Pour rappel, il s'agit d'un vecteur ligne dont les éléments correspondent aux probabilités de se situer initialement dans les phases correspondantes. La somme des éléments de ce vecteur doit donc être égale à 1.

Lors d'un renouvellement, les transitions associées sont déterminées par la matrice définie par $t \cdot \tau$. Nous pouvons dès lors écrire un générateur G pour le processus de Markov $\{(N_t, J_t), t \in \mathbb{R}_0^+\}$ à l'aide de la matrice définie par blocs suivante :

$$G = \begin{bmatrix} T & t \cdot \tau & & & \\ & T & t \cdot \tau & & \\ & & T & t \cdot \tau & \\ & & & \ddots & \ddots \end{bmatrix} \tag{6.11}$$

6.5 Description du Markovian Arrival Process

Reprenons à présent la définition du processus de renouvellement de type phase et son générateur G introduit en (6.11) afin d'introduire le MAP (aussi appelé Markovian Arrival Process).

Nous avons vu que lorsque nous quittions une phase lors d'une absorption en phase 0, la nouvelle phase est alors systématiquement choisie à l'aide du vecteur de probabilités initial caractérisé par le vecteur τ. Dès lors, indépendamment de la phase dans laquelle le renouvellement a été effectué, nous sommes assurés que la nouvelle phase est choisie de manière indépendante des précédentes. De cette manière, les intervalles entre les renouvellements demeurent indépendants et identiquement distribués. Cependant, nous avons parfois besoin de modèles où on observe une forte corrélation entre les temps inter-renouvellements. Cela nous pousse à établir un nouveau modèle où l'état du système après un renouvellement n'est pas déterminé de manière indépendante à chaque renouvellement.

Jusque maintenant, les transitions associées à un renouvellement étaient déterminées par la matrice $t \cdot \tau$. Les vecteurs lignes de cette matrice sont les suivants :

$$\begin{pmatrix} t_1 \cdot \tau \\ t_2 \cdot \tau \\ \vdots \\ t_n \cdot \tau \end{pmatrix} \tag{6.12}$$

Cela signifie que le seul paramètre qui diffère entre les lignes est le scalaire t_i, impliquant que la nouvelle phase après un renouvellement est déterminée indépendamment de la phase qui précédait le renouvellement.

Afin que ce choix devienne dépendant de la phase précédent le renouvellement, transformons cette matrice ainsi :

$$\begin{pmatrix} t_1 \cdot \tau_1 \\ t_2 \cdot \tau_2 \\ \vdots \\ t_n \cdot \tau_n \end{pmatrix} \tag{6.13}$$

où il est nécessaire d'imposer que $\tau_i \mathbf{1} = 1$ pour $i = 1, \ldots, n$ car chaque τ_i correspond à un vecteur de probabilités. Chacun de ces τ_i est associé à la phase i qui précédait le renouvellement.

Notons cette matrice nouvellement créée D_1, et notons $D_0 := T$. Le générateur G défini en (6.11) devient alors

$$G = \begin{bmatrix} D_0 & D_1 & & & \\ & D_0 & D_1 & & \\ & & D_0 & D_1 & \\ & & & \ddots & \ddots \end{bmatrix} \tag{6.14}$$

Un processus de Markov dont le générateur correspond à cette matrice est appelé Markovian Arrival Process. Contrairement aux processus de renouvellement de type phase, le MAP n'est pas un processus de renouvellement. Effectivement, comme nous l'avons défini, un processus de renouvellement est un processus dont la suite des temps inter-événements forme une suite de variables aléatoires indépendantes et identiquement distribuées. Cependant, ces variables aléatoires sont dépendantes et ne sont pas identiquement distribuées dans le cas du MAP. De fait, la distribution de la phase choisie suite à un renouvellement dépend de la phase qui a précédé le renouvellement.

Chapitre 7

Simulation du Markovian Arrival Process

Ce chapitre a pour but de proposer un modèle de simulation de Markovian Arrival Process sur base de la description effectuée dans le chapitre 6.

Une fois le modèle décrit, nous nous en servons dans le cadre d'une simulation d'un protocole de diffusion de rumeur en temps continu. Dans cette optique, le MAP est utilisé afin de définir les temps séparant chaque interaction entre les individus.

7.1 Présentation du modèle

Décrivons un modèle possible pour la simulation d'un MAP. Après avoir établi les objectifs de la simulation, nous effectuons un bref rappel sur le générateur infinitésimal associé à des chaînes de Markov en temps continu. Ensuite, nous procédons à la description du modèle conceptuel et du modèle de spécification. Finalement nous abordons l'implémentation et la validation de la simulation, afin de se convaincre que notre simulation correspond à celle escomptée.

7.1.1 Objectifs

La simulation a pour but d'effectuer une succession de renouvellements associés à un Markovian Arrival Process. Ce MAP est défini à l'aide des deux matrices D_0 et D_1, introduites dans la section 6.5.

A l'issue de la simulation, nous souhaitons connaître les temps T_i, pour $i \in \mathbb{N}_0$. Le temps T_i correspond à l'instant auquel le i-ème renouvellement a lieu. Ces temps nous permettent de déduire la durée X_i, introduits en (6.1) ; séparant le i-ème renouvellement du précédent, où X_1 est la durée séparant le premier renouvellement de l'instant 0.

7.1.2 Interprétation du générateur infinitésimal

Comme notre but est de simuler des renouvellements sur base des matrices D_0 et D_1 formant le générateur infinitésimal G introduit en (6.14), il est pertinent de rappeler ce qu'est un générateur infinitésimal.

Soit une chaîne de Markov en temps continu défini sur l'ensemble des états $\{0, \ldots, M\}$. Notons τ_i la variable aléatoire du temps passé dans l'état i avant de se déplacer vers un autre état, avec $i \in \{0, \ldots, M\}$. À ce processus de Markov est associé un générateur infinitésimal Q. Ce générateur prend la forme suivante :

$$Q = \begin{bmatrix} Q_{00} & Q_{01} & \cdots \\ Q_{10} & Q_{11} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{7.1}$$

où

$$Q_{ii} = -\sum_{\substack{j=0 \\ j \neq i}}^{M} Q_{ij} \qquad \forall i \in \{0, \ldots, M\} \tag{7.2}$$

Une propriété du processus de Markov en temps continu est que la variable τ_i est distribuée comme une variable aléatoire exponentielle. Le paramètre de la distribution exponentielle τ_i est $-Q_{ii}$. Autrement dit, l'élément Q_{ii} de la diagonale principale permet de déduire le temps moyen passé dans un état i avant une transition d'état.

Ainsi, l'espérance de la variable τ_i est exprimée de la manière suivante :

$$E[\tau_i] = \frac{1}{-Q_{ii}} \tag{7.3}$$

Comme τ_i est distribué comme une exponentielle de paramètre $-Q_{ii}$, nous pouvons également en déduire que

$$P[\tau_i \leq t] = 1 - e^{Q_{ii}t} \qquad \forall t > 0 \tag{7.4}$$

Les éléments Q_{ii} sur la diagonale principale du générateur Q nous permettent dès lors de déduire le temps moyen passé à chaque visite de l'état i. Les autres éléments Q_{ij} du générateur nous donnent les taux de transitions de l'état i vers un état j. Il s'agit donc du nombre moyen de fois que le processus passe de l'état i à l'état j par unité de temps passé dans l'état i.

Lors d'un changement d'état, la probabilité de quitter l'état i pour passer à l'état j est donnée par l'élément P_{ij} de la matrice P. Ces éléments P_{ij} sont calculés depuis le générateur Q de la manière suivante :

$$\begin{aligned} P_{ij} &= \frac{Q_{ij}}{-Q_{ii}} & i \neq j \\ P_{ij} &= 0 & i = j \end{aligned} \tag{7.5}$$

Lorsque nous souhaitons utiliser le générateur Q afin de simuler une chaîne de Markov en temps continu, une des manières de procéder est celle-ci, si nous partons d'un état i :

1. déterminer le temps avant le prochain changement d'état, déterminé à l'aide de la variable τ_i distribuée comme une variable exponentielle de paramètre $-Q_{ii}$ où i est l'état actuel

2. déterminer le nouvel état à partir de la matrice P, sachant que la probabilité P_{ij} correspond à la probabilité de transiter vers l'état j depuis l'état i lors d'un changement d'état

3. retour à la première étape, en considérant l'état actuel comme l'état déterminé à la deuxième étape.

7.1.3 Modèle conceptuel

Déterminons la structure à employer afin d'approcher au mieux la simulation.

L'état du système à un instant t pour $t \in \mathbb{R}^+$ peut être résumé à l'aide de trois informations. La première est l'instant t correspondant au temps écoulé depuis le début de la simulation. La deuxième information est J_t qui est associée à la phase

dans laquelle se trouve le système à l'instant t avec $J_t \in \{0, \ldots, n\}$, où n est le nombre de phases en plus de la phase 0 absorbante. Finalement, la troisième variable est $N_t \in \mathbb{N}_0$. Cette variable nous donne le nombre de renouvellements ayant pris place dans l'intervalle de temps $(0, t)$.

Les changements de phase et les renouvellements sont déterminés à l'aide des deux matrices D_0 et D_1, formant le générateur infinitésimal G suivant, introduit en (6.14) :

$$G = \begin{bmatrix} D_0 & D_1 & & & \\ & D_0 & D_1 & & \\ & & D_0 & D_1 & \\ & & & \ddots & \ddots \end{bmatrix} \tag{7.6}$$

Dans le modèle proposé ici, deux matrices permettent de déterminer la phase vers laquelle nous transitons lorsqu'un changement de phase a lieu. Une matrice P pour les changements de phase qui ne sont pas liés à un renouvellement, et une matrice P' déterminant la nouvelle phase succédant un renouvellement.

La matrice P correspond à une matrice de dimension $(n+1) \times (n+1)$. Elle est construite à partir de la matrice Q introduite en (6.3) :

$$\begin{aligned} P_{ij} &= \frac{Q_{ij}}{-Q_{ii}} && i \neq j \wedge i \neq 0 \\ P_{ij} &= 0 && i = j \vee i = 0 \end{aligned} \tag{7.7}$$

avec

$$Q = \begin{bmatrix} 0 & 0 \\ t & D_0 \end{bmatrix} \tag{7.8}$$

où t est défini en (6.4).

Les éléments P_{ij} donnent la probabilité de transiter d'une phase i à une phase j lors d'un changement de phase non-associé à un renouvellement. Ces éléments respectent la contrainte suivante :

$$\sum_{j=0}^{n} P_{ij} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{7.9}$$

La phase 0 étant absorbante, la première ligne de la matrice P n'a aucune importance, celle-ci n'étant jamais sollicitée. Effectivement, dès que nous arrivons dans la phase absorbante, un renouvellement est effectué et une nouvelle phase est choisie indépendamment de P.

Lors d'un renouvellement, c'est-à-dire lors d'une transition dans la phase absorbante, la nouvelle phase est déterminée à l'aide de la matrice P'. La matrice P' est de dimension $n \times n$ et est construite à partir de la matrice D_1 :

$$P'_{ij} = \frac{D_{1ij}}{\sum_{k=1}^{n} D_{1ik}} \tag{7.10}$$

Les éléments de la i-ème ligne de P' nous donnent chacun la probabilité d'arriver dans la phase correspondant à leur colonne si la phase qui a précédé le renouvellement était la phase i. L'élément P'_{ij} nous donne donc la probabilité de transiter vers la phase j sachant que la phase précédent le renouvellement était la phase i.

Initialement, nous commençons à l'instant $t = 0$, en phase 1. Cependant, il est envisageable de déterminer la phase initiale depuis un vecteur de probabilités initial.

L'objectif principal de la simulation est d'obtenir une liste de temps T_i correspondant aux instants associés aux renouvellements. Dès lors, nous employons une liste de réels pour enregistrer ces temps T_i.

7.1.4 Modèle de spécifications

Décrivons à présent le comportement des variables principales sur base de la structure établie dans le modèle conceptuel.

Initialement, la simulation commence à l'instant $t = 0$. Au fur et à mesure de la simulation, la variable t va croître. A chaque changement de phase, la variable t va être augmentée de la durée écoulée depuis le dernier changement de phase.

La variable J_t vaut initialement 1. À chaque changement de phase, cette variable évolue en fonction de la nouvelle phase adoptée. La nouvelle valeur de J_t est alors déterminée en fonction de la matrice P. Cette valeur est un naturel compris entre 0 et n, où n est le nombre de phases en plus de la phase absorbante. Lors d'un renouvellement, la matrice P' est utilisée afin de déterminer la nouvelle phase, et par conséquent la nouvelle valeur de J_t.

La variable N_t vaut initialement 0. À chaque renouvellement, la valeur de cette variable est incrémentée de 1. Pour rappel, un renouvellement est effectué lorsqu'une transition vers une phase absorbante est effectuée.

A chaque renouvellement, nous prenons également soin de mettre à jour la liste des temps T_i afin de sauvegarder les instants associés à chaque renouvellement. Ainsi, les renouvellements prennent place tour à tour jusqu'à ce qu'une condition d'arrêt soit respectée. Notre condition d'arrêt correspond ici à une valeur maximale pour la

variable t au-delà de laquelle la simulation prend fin.

La valeur de N_t à un instant t peut être directement calculée depuis la liste des temps T_i. Effectivement, la longueur de la liste de ces temps correspond au nombre de renouvellements ayant eu lieu dans l'intervalle de temps $(0, t)$, à savoir la valeur de N_t à cet instant t.

7.1.5 Implémentation et validation

L'implémentation effectuée dans le cadre de ce mémoire est réalisée à l'aide du langage Python.

Après avoir initialisé la matrice D_0 et D_1, nous calculons la valeur de la matrice P sur base de D_0 en respectant l'expression (7.7). Nous calculons également la matrice P'. La variable t est initialisée à 0, et J_t à 1. Initialement, aucun renouvellement n'a eu lieu, la liste des instants associés aux renouvellements T_i est donc vide.

Nous rentrons ensuite dans la boucle principale :

1. nous calculons le temps requis pour un changement de phase. Pour calculer ce temps, nous générons un nombre aléatoire distribué comme une exponentielle dont le paramètre est déterminé en fonction de la diagonale principale de D_0

2. nous mettons à jour la variable t en lui rajoutant la valeur calculée à l'étape précédente

3. a partir de la matrice P, nous déterminons la nouvelle phase. Pour ce faire, nous regardons dans P la ligne associée à la phase actuelle. Les éléments de cette ligne déterminent la probabilité de transiter vers chaque phase. Nous générons donc un nombre uniformément entre 0 et 1, et en fonction de sa valeur nous déterminons vers quelle phase transiter

4. si la nouvelle phase est la phase absorbante, je sauvegarde la valeur actuelle de t dans la liste des instants T_i, et je détermine la phase succédant le renouvellement à l'aide de la matrice P'. A l'instar de l'utilisation de la matrice P pour déterminer la nouvelle phase, nous générons aléatoirement un nombre distribué comme une variable uniforme définie entre 0 et 1. Nous regardons la i-ème ligne de P' associée à la phase i qui a précédé le renouvellement, et déterminons la nouvelle phase sur base des probabilités P'_{ij}.
   Par exemple, considérons le cas où nous nous trouvions dans la phase 2 avant un renouvellement sur une situation où D_0 est de dimension 3×3. Soit le vecteur $[P'_{21}, P'_{22}, P'_{23}]$ défini comme $[0.3, 0.2, 0.5]$. Si le nombre aléatoire généré uniformément est 0.4, cela signifie que la nouvelle phase sera phase 2, étant

donné que $P'_{21} < 0.4 < P'_{21} + P'_{22}$

5. si le temps t n'est pas supérieur à la valeur associée à la condition d'arrêt, on retourne à l'étape 1 de la boucle.

A l'issue de la simulation, nous sommes en possession de la liste associée aux instants des différents renouvellements, notre objectif étant alors accompli.

Dans l'optique de valider notre simulation, il peut être pertinent de garder en mémoire des informations supplémentaires comme les instants des changements de phases ainsi que les phases associées à chaque changement de phase.

Si la simulation se comporte comme nous le désirons, nous sommes alors en mesure d'observer un temps moyen avant de quitter les différentes phases proche de la valeur théorique $\frac{-1}{D_{0ii}}$. Cette valeur théorique correspond pour chaque phase i avec $i \in \{1, \ldots, n\}$ à l'espérance de l'exponentielle de paramètre D_{0ii} et a été introduite en (7.3).

Un outil supplémentaire nous permettant de valider la simulation est d'observer que la proportion des transitions effectuées de la phase i et la phase j correspond bien à celle suggérée par les matrices P et P'.

7.1.6 Exemple

A titre d'exemple, simulons un MAP défini par les matrices D_0 et D_1 suivantes :

$$D_0 = \begin{bmatrix} -0.6 & 0.06 \\ 0.01 & -0.1 \end{bmatrix} \tag{7.11}$$

$$D_1 = \begin{bmatrix} 0.48 & 0.06 \\ 0.01 & 0.08 \end{bmatrix} \tag{7.12}$$

Nous observons 2 phases : l'une associée à des renouvellements plus fréquents et l'une à des renouvellements moins fréquents. Les éléments de la diagonale principale de D_0 nous permettent de savoir combien de temps en moyenne il faut attendre pour quitter les phases. Nous nous attendons ainsi à devoir attendre en moyenne 1/0.6 unité de temps avant de quitter la phase 1, et en moyenne 1/0.1 unité de temps pour quitter la phase 2.

La matrice Q introduite en (7.8) est la suivante :

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0.54 & -0.6 & 0.06 \\ 0.09 & 0.01 & -0.1 \end{bmatrix} \tag{7.13}$$

Calculons à présent les valeurs de P et P' afin de déterminer le comportement du système lors d'un changement de phase.

$$P = \begin{bmatrix} 0 & 0 & 0 \\ \frac{Q_{10}}{-Q_{11}} & 0 & \frac{Q_{12}}{-Q_{11}} \\ \frac{Q_{20}}{-Q_{22}} & \frac{Q_{21}}{-Q_{22}} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0 & 0.1 \\ 0.9 & 0.1 & 0 \end{bmatrix} \tag{7.14}$$

$$P' = \begin{bmatrix} \frac{D_{111}}{D_{111}+D_{112}} & \frac{D_{112}}{D_{111}+D_{112}} \\ \frac{D_{121}}{D_{121}+D_{122}} & \frac{D_{122}}{D_{121}+D_{122}} \end{bmatrix} = \begin{bmatrix} 0.8889 & 0.1112 \\ 0.1112 & 0.8889 \end{bmatrix} \tag{7.15}$$

La matrice P nous indique qu'à l'issue d'un changement de phase, nous avons 90% de chance d'aller dans la phase 0 absorbante et d'avoir un renouvellement. Les 10% restants sont associés à un changement de la phase 1 à la phase 2 ou inversement. La matrice P' nous apprend qu'après un renouvellement, nous avons 88.89% de chance de demeurer dans la phase qui a précédé le renouvellement.

Le comportement de ce processus est donc un comportement où nous devrions avoir des intervalles de temps constitués de "rafales" de renouvellements, associés à la phase 1. Ces rafales devraient être séparées de moments plus calmes où le temps moyen entre les renouvellements est 6 fois plus long, lors d'un séjour en phase 2.

Établissons la limite $t > 100$ comme condition d'arrêt à la simulation. La figure 7.1 représente l'évolution de N_t sur le temps. Les portions en rouge de la courbe correspondent aux intervalles de temps passés dans la phase 1, et les portions en bleu aux intervalles associés à la phase 2. Nous remarquons qu'effectivement, les renouvellements semblent s'enchaîner rapidement lors d'un séjour dans la phase 1 et semblent être beaucoup plus espacés lorsque nous nous trouvons en phase 2.

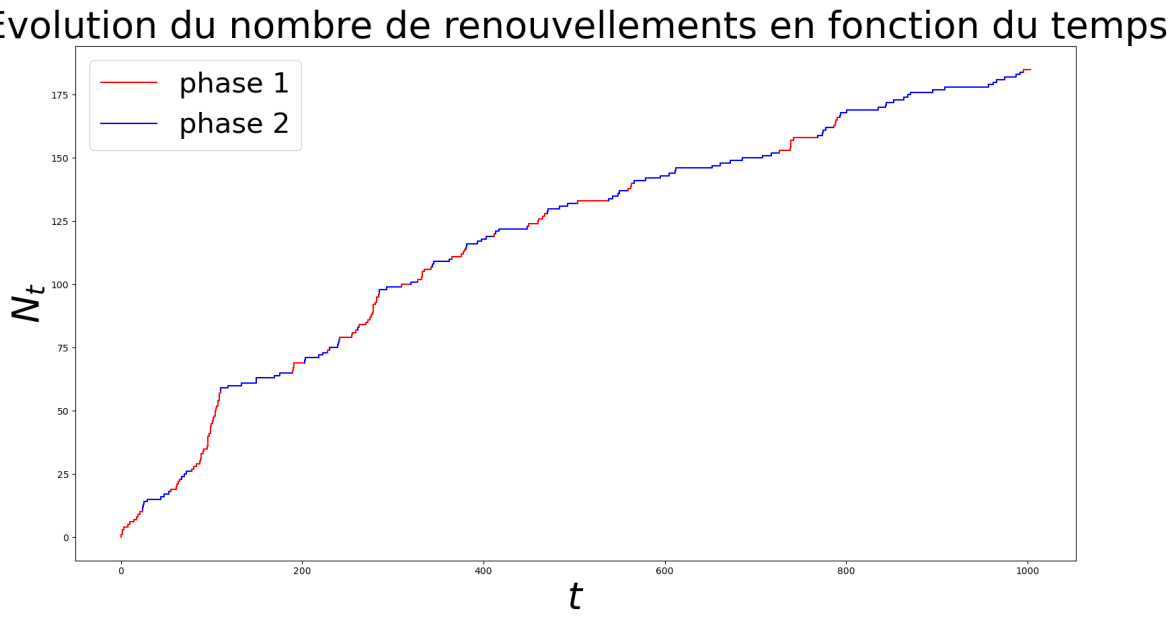

FIGURE 7.1 – Évolution du nombre de renouvellements N_t au fil du temps

Sur cette simulation, le temps moyen empirique passé en phase 1 avant un changement de phase est de 1.535 unité de temps et de 9.03 pour la phase 2. Les valeurs théoriques de ces temps moyens sont respectivement de 1.667 et de 10.

La figure 7.2 représente les différentes proportions observées vis-à-vis des transitions de phases suite à un renouvellement. Parmi les renouvellements succédant un séjour dans la phase 1, nous observons 88% de transitions où nous demeurons dans la phase 1 et 12% où nous transitons dans la phase 2. En revanche, pour les renouvellements qui succèdent à la phase 2, nous observons 84% de cas où la phase 2 est à nouveau choisie suite au renouvellement et 16% de cas où on passe à la phase 1. Dans les deux cas, la probabilité théorique de demeurer dans la phase qui a précédé le renouvellement est de 88.89%.

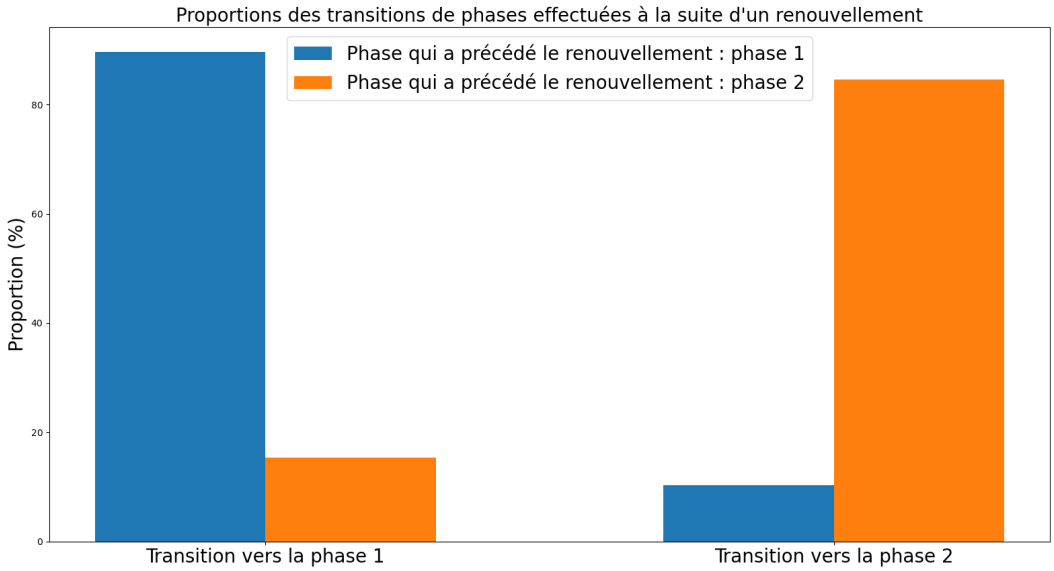

FIGURE 7.2 – Proportions associées aux transitions de phases effectuées suite à un renouvellement

7.2 Application au protocole de diffusion de rumeur

Regardons à présent comment adapter le Markovian Arrival Process au protocole de diffusion de rumeur en temps continu introduit dans le chapitre 5. Jusque maintenant, les interactions entre les individus du système étaient rythmées par un processus de Poisson de paramètre λ. Les temps qui séparaient les différentes interactions étaient donc indépendants et identiquement distribués. Cependant, il peut être adéquate dans certaines situations d'employer un modèle où les interactions sont rythmées d'une manière différente. C'est pourquoi le MAP a été introduit dans le cadre de ce mémoire.

Le MAP a l'avantage de retirer la propriété d'indépendance et de l'identique distribution des différents temps inter-renouvellements. Avec une estimation adéquat des matrices D_0 et D_1, comme proposée en [2], il permet une modélisation beaucoup plus complexe du rythme séparant les interactions. Si par exemple nous souhaitons observer deux comportements : l'un où les interactions se succèdent et l'un où les interactions sont beaucoup plus espacées, un choix similaire à celui présent dans l'exemple de la sous-section 7.1.6 peut être effectué.

Pour appliquer le MAP au protocole en diffusion de rumeur, nous partons du modèle

de simulation introduit en temps discret dans le chapitre 5. Le temps t n'est ici plus incrémenté de 1 à chaque interaction, étant donné que nous sommes en temps continu et non plus dans le cadre discret. A la place, un Markovian Arrival Process est simulé. A chaque renouvellement, une interaction est effectuée entre deux individus. Le choix des individus et l'issue de l'interaction sont inchangés par rapport au modèle vu en temps discret.

L'unique différence est que la manière avec laquelle le temps t augmente avec chaque renouvellement. De cette manière, à l'issue de la simulation, nous sommes en mesure de déterminer l'instant associé à chaque interaction. Le temps de convergence T_n devient le temps requis pour atteindre l'état de convergence, c'est-à-dire l'état où la rumeur s'est propagée à chaque individu.

Sur la figure 7.3 est représentée une simulation de l'utilisation du MAP dans le cadre du protocole de diffusion de rumeur en temps continu. Le nombre d'individus dans le système est de 100. Les matrices D_0 et D_1 sont les mêmes que celles utilisées dans l'exemple de la sous-section 7.1.6, à savoir :

$$D_0 = \begin{bmatrix} -0.6 & 0.06 \\ 0.01 & -0.1 \end{bmatrix} \tag{7.16}$$

$$D_1 = \begin{bmatrix} 0.48 & 0.06 \\ 0.01 & 0.08 \end{bmatrix} \tag{7.17}$$

En abscisse est représenté le temps, mais qui ne correspond plus au nombre d'interactions ayant eu lieu depuis le début comme en temps discret. En ordonnée est représenté Z_t, à savoir le nombre d'individus connaissant la rumeur à un instant t.

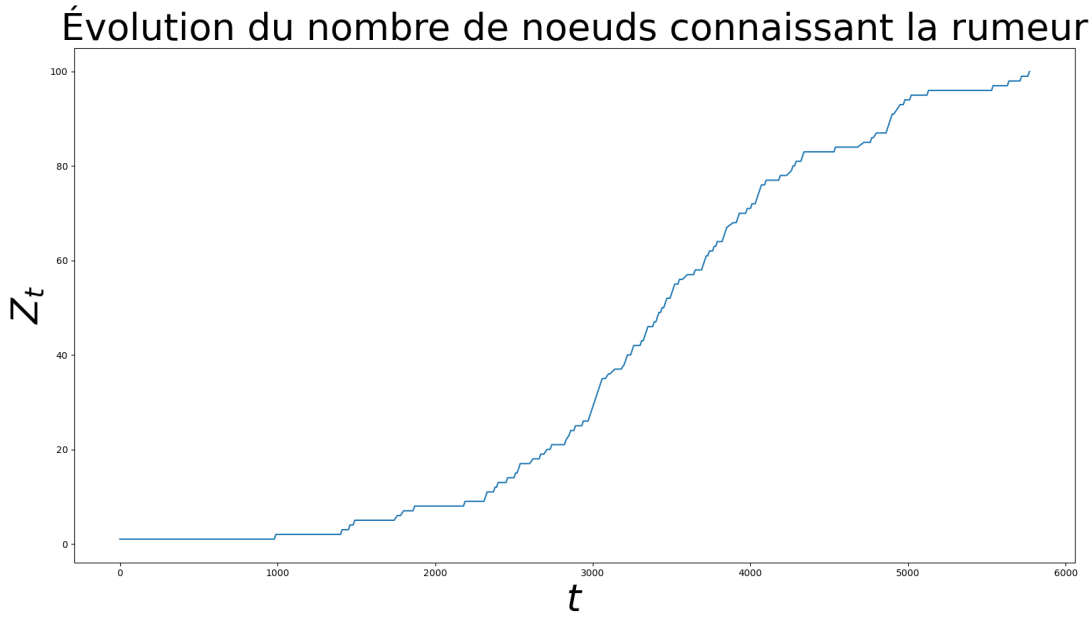

FIGURE 7.3 – Évolution du nombre d'individus connaissant la rumeur en fonction du temps

Chapitre 8

Conclusion

Ce mémoire est principalement axé autour de la description et de la modélisation du protocole de diffusion de rumeur en temps discret et continu. Après une brève classification des protocoles de population, nous avons introduit le protocole de diffusion de rumeur.

Sur base des travaux d'Yves Mocquard, une modélisation du protocole de diffusion de rumeur en temps discret a été proposée. Nous avons discuté ensuite du comportement du temps de convergence T_n ainsi que du calcul de la distribution qui lui est associée. La modélisation adaptée au temps continu a ensuite été abordée.

Une fois le protocole de diffusion de rumeur introduit, nous avons proposé un modèle de simulation de ce protocole en temps discret, ainsi que plusieurs méthodes potentielles pour valider les simulations effectuées.

Afin de proposer une approche plus complexe du protocole de diffusion de rumeur, nous avons souhaité modifier la distribution déterminant les temps entre les interactions. Pour ce faire, nous avons introduit les processus de renouvellement ainsi que le Markovian Arrival Process, et avons proposé un modèle de simulation pour le MAP. Désormais capables de simuler un Markovian Arrival Process, nous l'avons appliqué au protocole de diffusion de rumeur afin de proposer une alternative au processus de Poisson.

Bibliographie

[1] Berger N., Borgs C., Chayes J. T., Saberi A. On the spread of viruses on the internet. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithm (SODA)*, 2005.

[2] Metzner P., Dittmer E., Jahnke T., Schütte CH. Generator estimation of markov jump processes. *Journal of Computational Physics*, 227, 2007.

[3] Angluin D., Aspnes J., Eisenstat D. Fast computation by population protocols with a leader. *Distributed Computing*, 21, 2008.

[4] Demers A., Gealy M., Greene D., Hauser C., Irish W., Larson J., Shenker S., Sturgis H., Swinehart D., Terry D. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1989.

[5] Harchol-Balter M., Leighton T., Lewin D. Resource discovery in distributed networks. In *Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing (PODC)*, 1999.

[6] Mocquard Y., Sericola B., Anceaume E. Probabilistic analysis of rumor spreading time. *INFORMS Journal on Computing*, 32, 2020.

[7] Kermack W. O., McKendrick A. G. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society A*, 115, 1927.

[8] Kempe D., Dobra A., Gehrke J. Gossip-based computation of aggregate information. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, 2003.

[9] Bawa M., Garcia-Molina H., Gionis A., Motwani R. Estimating aggregates on a peer-to-peer network. Technical report, Stanford InfoLab, 2003.

[10] Latouche G. , Ramaswami V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics, 1999.

[11] Mocquard Y. *Analyse probabiliste de protocoles de population*. PhD thesis, Université de Rennes 1, 2018.