

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider, Biskra  
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie  
Département de Mathématiques



Mémoire présenté pour obtenir le diplôme de

Master en **Mathématiques Appliquées**

Option : Statistique

Par Melle. **DRIHEM Sofia**

**Titre :**

Analyse statistiques d'un système d'attente avec feedback  
de clients

Devant le Jury :

Mme. Benameur Sana	MCA	U. Biskra	Présidente
Mr. Cherfaoui Mouloud	Pr	U. Biskra	Rapporteur
Mme. Benbrika Ghozlane	MAA	U. Biskra	Examinatrice

**Soutenu Publiquement le 27/06/2022**

## *Dédicace*

*Je* dédier ce modeste travail

À mes chers parents :

Ma mère, symbole de tendresse, qui s'est sacrifiée pour mon bonheur  
et ma réussite.

Mon père, École de mon enfance, qui a veillé à me protéger, m'aider  
et m'encourager.

À Ceux qu'ont été près de moi, qui ont grandi avec moi dans le bon et le  
mal, mon frère et ma soeur.

À mes amies : Marwa, Lamia, Hind, Chourouk, Hadjer.

# Remerciements

*Je* tiens à remercier sincèrement le tout puissant ALLAH qui m'a aidé et m'a donné la santé, la patience et le courage durant mes longues années d'étude et la force pour finir ce travail.

*Je* tiens tout particulièrement à remercier mon encadreur Pr. Cherfaoui Mouloud pour son suivi et l'aide qu'il m'a apporté pour l'élaboration et pour ses précieux conseils et ses aides durant toute la période du travail de ce mémoire.

*Je* remercie également tous les enseignants et employés de la Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie, en particulier ceux du Département de Mathématiques.

*Je* n'oublie pas mes parents pour leur contribution, leur soutien et leur patience. Enfin, j'adresse mes plus sincères remerciements à toute ma famille, spécialement mes chers parents et mes frères et à tous, pour tout.

# Résumé du mémoire

**Résumé :** Dans ce mémoire nous avons réalisé via l'approche simulation à événements discrets une analyse détaillée d'un système d'attente  $GI/GI/1$  avec Bernoulli feedback de clients dont la probabilité de Feedback dépend du nombre de clients dans le système. Les résultats de simulation obtenus (numérique et graphiques) mettent en évidence principalement l'effet de la distribution des durées des inter-arrivées, la distribution des durées de service, la probabilité de Feedback et l'intensité du trafic sur les caractéristiques stationnaire du système en question.

**Mots clés :** Files d'attente, processus (non)Markovien, feedback, mesures de performance, simulation.

**Abstract :** In this thesis, we have carried out via discrete events simulation approach a detailed analysis of a  $GI/GI/1$  queuing system with Bernoulli customer feedback whose feedback probability depends on the number of customers in the system. The simulation obtained results (numerical and graphical) mainly highlight the effect of the distribution of inter-arrivals times, the distribution of service times, the probability of Feedback and the traffic intensity on the stationary characteristics of the system in question.

**Keywords :** Queues, (non)Markovian process, feedback, performance measures, simulation.

# Table des matières

Dédicace	i
Remerciements	ii
Résumé du mémoire	iii
Table des matières	iv
Table des figures	vii
Liste des tableaux	viii
Introduction	1
<b>1 Simulation et génération des nombres pseudo-aléatoires</b>	<b>3</b>
Introduction . . . . .	3
1.1 Simulation . . . . .	3
1.1.1 Terminologies . . . . .	4
1.1.2 Étapes de la simulation . . . . .	6
1.1.3 Simulation à événements discrets . . . . .	8

1.1.4	Simulation de Monte Carlo . . . . .	10
1.2	Génération des nombres pseudo-aléatoires . . . . .	11
1.2.1	Méthodes de génération des nombres pseudo-aléatoires . . . . .	11
1.2.2	Quelques tests statistiques . . . . .	18
	Conclusion . . . . .	21
<b>2</b>	<b>Systèmes d'attente Markoviens avec Bernoulli feedback</b>	<b>22</b>
	Introduction . . . . .	22
2.1	Définition des files d'attente . . . . .	22
2.2	Notation de Kendall-Lee . . . . .	23
2.3	Mesures de performance d'une file d'attente . . . . .	27
2.4	Files d'attente Markoviennes avec Feedback . . . . .	28
2.4.1	Système d'attente $M/M/1$ avec Bernoulli feedback . . . . .	28
2.4.2	Système d'attente $M/M/m$ avec Bernoulli feedback . . . . .	31
	Conclusion . . . . .	34
<b>3</b>	<b>Evaluation de performance d'un système d'attente avec feedback</b>	<b>35</b>
	Introduction . . . . .	35
3.1	Plan de simulation . . . . .	36
3.2	Simulation du système d'attente $GI/GI/1$ . . . . .	38
3.3	Validation du modèle de simulation . . . . .	43
3.4	Effet de l'intensité de $\rho$ et de la probabilité $\beta_n$ . . . . .	44
3.4.1	Résultats de simulation . . . . .	45
3.4.2	Analyse et interprétation des résultats . . . . .	48

3.5	Effet des distributions des processus d'arrivées et de service . . . .	50
3.5.1	Résultats de simulation . . . . .	51
3.5.2	Analyse et interprétation des résultats . . . . .	52
	Conclusion . . . . .	53
	<b>Conclusion générale</b>	<b>54</b>
	<b>Bibliographie</b>	<b>56</b>
	<b>Annexe A : Simulation de lois usuelles et la fonction <i>RANDOM</i></b>	<b>58</b>
3.6	Simulation de lois usuelles . . . . .	58
3.6.1	Loi Exponentielle . . . . .	58
3.6.2	Loi d'Erlang . . . . .	59
3.6.3	Loi de Weibull . . . . .	61
3.7	La fonction <i>RANDOM</i> sous Matlab . . . . .	61

# Table des figures

1.1	(A) : Processus de validation et (B) : Processus d'analyse et interprétation des résultats . . . . .	8
2.1	Représentation d'une file d'attente simple. . . . .	23
2.2	Representation d'une file d'attente $M/M/1$ avec Bernoulli feedback. . . . .	29
3.1	File d'attente $GI/GI/1$ avec feedback dépendant. . . . .	37
3.2	File d'attente $GI/GI/1$ avec feedback dépendant. . . . .	38
3.3	Programme MATLAB du Simulateur de la file $GI/GI/1$ avec feedback dépendant. . . . .	39
3.4	Variation de $L$ et $Var(\hat{L})$ en fonction du couple $(\beta_n, \rho)$ . . . . .	46
3.5	Variation de $P_0$ et $Var(\hat{P}_0)$ en fonction du couple $(\beta_n, \rho)$ . . . . .	47
3.6	Variation de $\bar{\beta}$ et $Var(\hat{\beta})$ en fonction du couple $(\beta_n, \rho)$ . . . . .	47
3.7	Variation de $N$ et $Var(\hat{N})$ en fonction du couple $(\beta_n, \rho)$ . . . . .	47



# Liste des tableaux

3.1	Performances théorique du système $M/M/1$ . . . . .	44
3.2	Résultats de simulation du système $M/M/1$ avec feedback dépendant.	45
3.3	Résultats de simulation du système $GI/GI/1$ avec feedback dépendant. . . . .	51

# Introduction

Lors de l'analyse d'un modèle stochastique, on se retrouve toujours confronter à des systèmes d'équations différentielles dont la résolution est complexe, ou possèdent des solutions qui ne sont pas facilement interprétables. Par ailleurs, on peut citer le degré de difficulté pour l'obtention de certaines caractéristiques dans quelques modèles, tels que les modèles de files d'attente non Markoviennes.

Toute la théorie de files d'attente est fondée sur celle des probabilités et des processus stochastiques, cela signifie qu'en théorie, on doit pouvoir prévoir l'évolution du comportement de ces systèmes avec certitude. Il s'agit pour cela de décrire leur fonctionnement finement et de considérer tous les paramètres pouvant influencer sur leurs comportements. Cependant, en pratique on ne dispose pas de toutes ces informations et même si on en disposait, on obtiendra un modèle tellement complexe qu'il serait quasiment impossible de le résoudre avec exactitude. Pour faire face ces situations nous faisons recours aux méthodes d'approximation au sens général et à la simulation en particulier.

Dans le présent mémoire nous avons considéré un exemple d'un processus stochastique dont le recours à la simulation, pour l'analyse de ce processus, est inévitable. En effet, dans le présent document nous avons considéré l'analyse de l'effet des paramètres de départ du système d'attente  $G/G/1$  avec feedback de clients sur les mesures de performance de ce dernier et ceci sous l'hypothèse que la probabilité

de feedback dépend du nombre de clients dans le système. Cependant, vu que les résultats analytique sur les mesures de performance de tels systèmes sont inexistant dans la littérature et sont très difficile à les concevoir, alors pour réaliser cette analyse nous avons eu recours à la technique de simulation à évènement discret.

Dans l'étude de simulation nous nous sommes concentrés particulièrement sur l'effet de trois paramètres, à savoir : l'intensité du trafic ( $= \{ \text{taux d'arrivées} \} / \{ \text{taux de service} \}$ ), la forme de la probabilité de feedback et les distributions des processus d'arrivées et celui du service. Par ailleurs, les résultats obtenu mettent en évidence d'effet de chacun de ces trois paramètres sur le nombre moyenne de clients dans le système, la probabilité que le système sois vide, la probabilité moyenne de feedback et le nombre moyen de service reçu par le même client.

Notre mémoire est structuré de la manière suivante :

- ▶ Le chapitre 1 aborde les notions de base sur la simulation à événements discrets et la génération des nombres pseudo-aléatoires.
- ▶ Le chapitre 2 comprend dans un premier une brève introduction sur les notations des systèmes de files d'attente. Par la suite, l'analyse de quelques systèmes d'attente Markoviennes avec Bernoulli feedback a été présenté.
- ▶ Le chapitre 3 sera consacré à l'analyse ainsi qu'à l'interprétation détaillée des performances des systèmes :  $M/GI/1$ ,  $GI/M/1$ ,  $GI/GI/1$ , et  $M/M/1$  avec feedback par l'approche simulation lorsque la probabilité de feedback dépend du nombre de clients dans le système.
- ▶ L'Annexe A comprend des méthodes de génération des nombres pseudo-aléatoires suivant certaines lois usuelles et la description de la fonction RANDOM du logiciel Matlab.

# Chapitre 1

## Simulation et génération des nombres pseudo-aléatoires

### Introduction

Lorsqu'une résolution mathématique exactes d'un problème n'est pas possible, on fait appel aux méthodes d'approximation. Entre autres, figure la méthode de la simulation qui est un outil puissant et idéal dans la majorité des sciences actuelles (Informatique, Physique, Chimie, Télécommunications,...) en plus est un outil de résolution numérique et une discipline de modélisation.

### 1.1 Simulation

Le terme "Simulation" est dérivé du mot latin *SIMULARE* qui veut dire : copier, feindre, faire paraître comme réelle une chose qui ne l'est pas. Nous conduit aussi vers le chemin qui nous permet de déduire les caractéristiques du fonctionnement d'un système réel. En d'autre terme, la simulation est l'étude du comportement

d'un système dynamique à travers un modèle (grâce à un modèle que l'on fait évoluer dans le temps), elle désigne un procédé selon lequel on exécute un programme informatique sur ordinateur en vue de simuler un phénomène ou une organisation particulière : Chaîne de montage en industrie, trafic routier dans une agglomération, bloc opératoire dans un hôpital, etc.

La simulation ne résout pas le problème posé en trouvant la solution, mais elle nous aide à prendre parmi plusieurs solutions la meilleure d'entre elles.

La simulation a pour rôle d'imiter l'évolution d'un système dans le temps et calculer ensuite ses caractéristiques. On a souvent recours à la simulation lorsque :

1. L'expérience sur le système est impossible ou dangereuse (exemple de réacteurs nucléaires).
2. Le système évolue rapidement de telle sorte qu'on ne peut pas l'observer avec précision (exemple de réactions chimiques).
3. Le système évolue très lentement (exemple de simulation du climat à long terme).
4. L'expérience est coûteuse financièrement.
5. Le système est en phase de conception.
6. On veut valider des résultats analytiques.

### 1.1.1 Terminologies

**Définition 1.1.1** *Un système est un ensemble doté d'une structure, d'un groupe d'éléments et de relations entre ceux-ci dans un environnement fermé. Il est affecté par les éventuels changements de son environnement.*

Dans le domaine de la simulation, plusieurs définitions ont été attribuées au

concept "Modèle", en voici celle donnée par l'AF CET (Association Française pour la Cybernétique économique et Technique)

**Définition 1.1.2** *"Un modèle est un schéma, c'est-à-dire une description mentale (intériorisée), ou figurée (diagramme, formules mathématiques, etc.) qui pour un champ de questions est pris comme représentation abstraite d'une classe de phénomènes, plus ou moins habilement dégagés de leur contextes par un observateur pour servir de support à l'investigation, et/ou la communication".*

*Le modèle est une représentation simplifiée d'un système réel ou imaginé exprimer sous forme verbale, graphique ou mathématique dans le but de l'étudier. Toutefois, il doit contenir assez de détails pour tirer des conclusions valables sur le système réel.*

**Définition 1.1.3** *Les variables d'état sont les informations nécessaires qui ont pour but de définir ce qui est en train de se passer dans un système.*

**Définition 1.1.4** *Les entités sont les éléments ou les objets du système qui subissent des opérations et se déplacent en générale dans le système (message dans un réseau,...). Une entité peut être active ou peut se résoudre d'une manière pacifique (passive), permanente ou temporaire.*

**Définition 1.1.5** *Les ressources sont des éléments qui exécutent des opérations, et généralement ne se déplacent pas dans le système (Machine, Unité Centrale,...). Mais on peut trouver des objets qui peuvent exécuter une opération tout en se déplaçant à l'intérieur du système (une machine transportant des palettes dans une usine, chariot transportant une pièce dans un atelier).*

**Définition 1.1.6** *Les attributs sont des variables identification qui caractérisent les entités ou les ressources. On distingue cependant deux types d'attributs :*

1. **Fixes** : *Contiennent les caractéristiques constantes de l'objet (durée de service, date d'arrivée dans le système, ...).*
2. **Variables** : *Contiennent les caractéristiques changeantes de l'objet (état d'une ressource, longueur de la file associée à une ressource,...).*

**Définition 1.1.7** *L'activité est un processus qui provoque un changement dans l'état du système, ce changement d'état est appelé événement. Les objets exécutent quelques opérations et dès que ces dernières seront initiées (ou terminées) à chaque événement, elles seront appelées des activités. Les activités utilisées dans la simulation à événement discret possèdent des durées.*

**Définition 1.1.8** *L'événement est la réalisation instantanée qui peut changer l'état du système. On distingue des événements internes au système (endogènes) et des événements externes (exogènes).*

**Définition 1.1.9** *Un processus est le rassemblement d'un certain nombre d'événements dans lequel ces derniers sont produits.*

**Définition 1.1.10** *Un simulateur est un programme contenant l'algorithme utilisé pour simuler le système étudié. Il est constitué d'un ensemble d'entités qui décrivent une composante du système réel.*

## 1.1.2 Étapes de la simulation

Pour concevoir un simulateur, on doit respecter les étapes suivantes :

- 1) **Formulation du problème** : Cette étape consiste à identifier et analyser le problème tout en déterminant ses composantes, ses relations et les frontières entre le système et son environnement.

- 2) **Fixation des objectifs** : Dans n'importe quel problème à simuler, il faut bien définir les objectifs visés par le projet de simulation. Ceci comprend :
  - les questions auxquelles on devra répondre.
  - Les sorties attendues.
  - Le temps requis ainsi que les coûts de l'étude, etc.
- 3) **Elaboration du modèle** : Il s'agit d'extraire un modèle qui est une abstraction du système réel, c'est-à-dire aussi fidèle que possible de ce dernier.
- 4) **Collecte de données** : Après la formulation d'un problème quelconque et après avoir identifié les objectifs, on fera appel à une collection de données qu'on adapte à des distributions de probabilité tout en faisant appel à des tests statistiques.
- 5) **Validation du modèle** : Il s'agit de traduire le modèle obtenu sous une forme acceptable par ordinateur et de s'assurer que le modèle s'exécute sans erreurs. Le meilleur moyen pour valider un modèle conceptuel est de comparer ses sorties avec celles du système réel. C'est dans cette étape qu'intervient les tests statistiques, c'est-à-dire pour pouvoir valider un modèle de simulation, on fait appel par exemple aux tests d'ajustements (*Khi-Deux* ou *Kolmogorov-Smirnov*), aux tests de comparaison (*test de Fisher* ou *test de Student*), ...etc. Le processus de validation d'un modèle peut être illustré par la figure 1.1.
- 6) **Exécution de la simulation** : L'exécution et le rassemblement des résultats d'un modèle permet de pouvoir le mettre à l'épreuve.
- 7) **Analyse et interprétation des résultats** : Une fois les résultats sont obtenus, le concepteur passe à l'interprétation de ces dernières dans le but de donner des propositions (solutions). Le choix de la meilleure solution devra être fait par l'analyse qui la justifiera dans une documentation qui est nécessaire pour différentes raisons :



- Aide à comprendre le fonctionnement du modèle.
- Facilite toute modification ou mise à jour du modèle.
- Permet de revoir toute les alternatives prises en considération, des critères de comparaisons qui ont été utilisés. Ceci nous aidera dans la prise de décision qui sera principalement basée sur les résultats fournis par la simulation et rapportés dans la documentation.

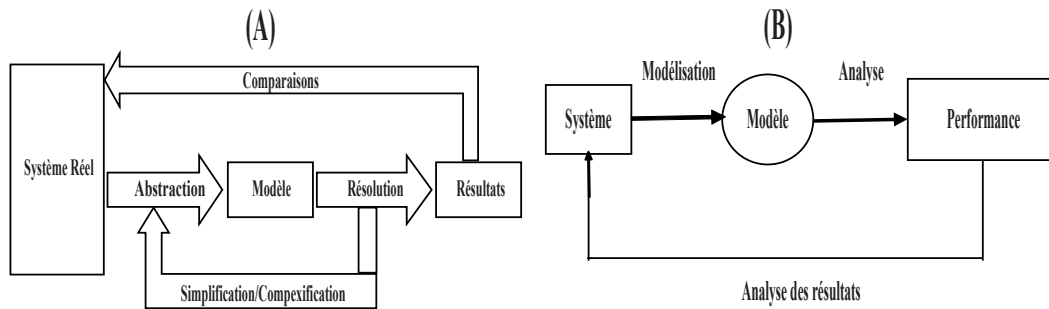


FIGURE 1.1 – (A) : Processus de validation et (B) : Processus d’analyse et interprétation des résultats

### 1.1.3 Simulation à événements discrets

La simulation à événements discrets est la représentation abstraite du système réel tel qu’il évolue dans le temps, c’est-à-dire se rapporte à la modélisation d’un système, dont l’état évolue au cours du temps selon une représentation dans laquelle les variables d’état changent instantanément à certaines dates précises, dans le but d’expliquer certains aspects du comportement de ce système.

#### Principales approches de la simulation à événements Discrets

Il existe quatre approches de la simulation à événements discrets et chaque approche est caractérisée par une méthode différente pour faire améliorer le système dans le temps :

1. **Approche activité (méthode a pas fixe)** : Lorsqu'on utilise cette méthode l'horloge est avancée par intervalle de temps discret et uniforme, et le système est examiné à chaque unité de temps, c'est-à-dire à chaque incrémentation, on vérifie et on cherche s'il y a des événements qui peuvent se produire ou des activités qui peuvent débiter pour pouvoir les traiter.
2. **Approche événements** : Cette méthode est économique en terme de temps de calcul, lorsqu'on l'utilise, on s'intéressera particulièrement à un événement, et l'horloge du temps sera avancé à chaque fois que ce dernier se produit . Ce qui veut dire qu'avant chaque déplacement de l'horloge, il est nécessaire de faire une vérification sur les événements qui nous sont réalisés juste avant, et s'il y a lieu on peut pas exécuter d'autres jusqu'à ce qu'on termine les précédents.
3. **Méthode des trois phases** : le principe de cette méthode consiste à :
  - (a) Déterminer le plus proche événement et de faire avancer l'horloge à ce point.
  - (b) Exécuter toute les activités B (Bound).
  - (c) Exécuter toute les activités C (Conditionnelle), dont les conditions sont satisfaites.
4. **Approche par processus** : La présence des séquences d'événements ou des activités similaires pour un type d'objets, défini sous forme de processus :
  - La description du fonctionnement du système complet par macro-représentation.
  - La gestion des conflits et la synchronisation entre processus par des règles d'interruption et de reprise.

### 1.1.4 Simulation de Monte Carlo

Cette méthode a été développée durant les années 40 par Von Neumann, Vlan et Fermi pour résoudre certains problèmes de désigne d'écrans anti-rayonnement. Elle est conçue à l'origine pour résoudre des problèmes déterministes et consiste à représenter un problème déterministe par un processus stochastique dont les distributions de probabilité satisfont les relations mathématiques du problème déterministe complexe [8, 7].

La première étape consiste à construire un modèle analytique qui représente l'investissement considéré (*exemple* : une équation pour la valeur présente). Dans la deuxième étape, l'analyste développe une distribution de probabilité pour chaque facteur qui est sujet à l'incertitude dans le modèle. Il peut utiliser des données historiques ou une approche subjective. À partir des distributions de probabilités pour chaque facteur dans le modèle, il génère au hasard une réponse tentative. La répétition de ce mode d'échantillonnage varie entre les différentes histoires bien que le modèle reste le même. Une question importante à connaître dans la simulation de Monté Carlo est le nombre d'essais (ou d'exécution) nécessaire pour avoir une réponse satisfaisante. La réponse est que le nombre d'essais doit être assez élevé pour atteindre le régime permanent.

On peut définir les conditions de régime permanent comme une situation dans laquelle le résultat d'essais consécutifs ne varie pas significativement. Les modèles analytiques et la simulation de Monté Carlo ont les mêmes problèmes, à savoir, la validité du modèle. Cependant, dans le cas de la simulation de Monté Carlo, la taille de l'échantillon doit être assez grande pour faire décroître la validité d'échantillonnage à un niveau acceptable. Les probabilités des événements doivent être basées sur les jugements de personnes impliquées dans le projet, utilisées avec

le jugement du décideur.

## 1.2 Génération des nombres pseudo-aléatoires

Toute simulation des phénomènes stochastique se fait par un moyen indirect, qui est la génération de nombres aléatoires, cette dernière consiste à obtenir des observations des variables aléatoires à partir de la distribution désirée, pour que ces observations puissent interpréter fidèlement la réalité du phénomène, il est nécessaire qu'elles aient la même loi de probabilité. L'objectif de la génération des nombres aléatoires est la construction d'une suite de nombres statistiquement indépendantes et réparties uniformément sur  $[0, 1]$  :

$$J_{n+1} = f(J_n), \quad (1.1)$$

où  $f$  est une fonction qui doit être choisie minutieusement et précisément, pour que la répartition des nombres  $J_n$  se rapproche de ce que donnerait le hasard. C'est à partir de là que la notion de nombres **pseudo-aléatoires** est apparue qui n'est autre qu'une suite de nombres aléatoires générée par un algorithme déterministe.

### 1.2.1 Méthodes de génération des nombres pseudo-aléatoires

#### Méthode congruentielle générale

Cette méthode, introduite par Lehmer en 1951, est une génération d'une suite de nombres par une formule tout à fait déterministe de manière à obtenir une suite qui semble aléatoire (indépendante et distribution uniforme dans l'intervalle de variation).

**Formule du générateur congruentiel**

Soit  $X_0$  un nombre entier positif. La suite de nombres est définie par :

$$x_{n+1} = (ax_n + c) \bmod [m]. \quad (1.2)$$

Avec :

- $m$  : module,  $m > 0$ .
- $a$  : multiplicateur,  $0 \leq a < m$
- $c$  : l'incrément,  $0 \leq c < m$ .

Dans le cas où :

1.  $c=0$  : le générateur congruentielle est dit linéaire.
2.  $c \neq 0$  : le générateur congruentielle est dit mixte.

**Propriétés et vocabulaire de la méthode congruentielle**

- $X_i < m$ .
- Le nombre maximum de valeurs possibles de la suite est  $m$ .
- Dès qu'un nombre est répété, toute la séquence recommence.
- $p$  dénote la période et le nombre de valeurs possibles de la suite.
- Si  $p = m$ , le générateur est dit de période maximum.

**Théorème 1.2.1** *Un générateur est de période maximum  $p = m$  si et seulement si les trois conditions suivantes sont vérifiées :*

- $c$  et  $m$  n'ont pas de diviseur commun,
- $a = 1$  modulo  $g$  pour tous les nombres premiers  $g$  diviseurs de  $m$  (Pour rappel,  $a = i$  modulo  $j$  si  $a$  peut s'exprimer comme  $a = i + kj$  où  $k$  est un entier positif ou nul).

**Remarque 1.2.1** *Pour obtenir des nombres pseudo aléatoires uniformément répartis sur  $[0,1]$ , il suffit de prendre :*

$$U_n = \frac{x_n}{m}. \tag{1.3}$$

Il s'agit de procréer (engendrer) une variable aléatoire  $X$  de fonction de densité  $f(x)$ , et de fonction de répartition  $\mathbb{F}(x)$ , suivant une loi uniforme sur  $[0,1]$  en se basant sur des techniques connues dont certaines s'appliquent à la génération de variables pseudo-aléatoires de distributions quelconques, tant dit que d'autres ne s'appliquent qu'aux distributions continues ou discrètes. Parmi les principales techniques de génération de nombres pseudo-aléatoires, on distingue :

1. **Méthode d'inversion** Si la fonction de répartition est explicitement connue, et la fonction de répartition inverse  $\mathbb{F}^{-1}$  existe. Dans ce cas, la méthode de la transformation inverse peut être utilisée comme suit :
  - a) Générer des variables  $Y \rightsquigarrow$  Uniforme  $[0,1]$  ;
  - b) Écrire  $Y$  en fonction de  $\mathbb{F}(X)$  ;
  - c) Déduire :  $x = \mathbb{F}^{-1}(y)$ .

**Remarque 1.2.2** La technique de transformation inverse est utilisée pour des distributions dont  $\mathbb{F}(x)$  est simple.

**Implémentation de la méthode d'inversion :**

- i) Cas discret : Générer  $n$  observations d'une variable aléatoire  $X$  qui suit une loi discrète.

$x_i$	$x_0$	$x_1$	$\dots$	$x_N$
$p_i$	$p_0$	$p_1$	$\dots$	$p_N$
$\mathbb{F}(x)$	$p_0$	$p_0 + p_1$	$\dots$	1

**Algorithme**

**Début**

**Pour**  $i$  allant de 1 à  $N$  **faire** :

Générer  $Y \rightsquigarrow U[0, 1]$  ;

Trouver  $n$  tel que  $\sum_{j=0}^n p_j \leq y < \sum_{j=0}^{n+1} p_j$

$x_i = x_n$  ;

**Fin Pour** ;

**Fin** ;

ii) Cas continue :

**Génération d'une loi uniforme sur  $[a, b]$  :**

La fonction de densité d'une variable aléatoire suivant une loi uniforme sur  $[a, b]$  est donnée par :

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b; \\ 0, & \text{ailleurs.} \end{cases} \quad (1.4)$$

La fonction de répartition est donnée par :

$$\mathbb{F}(x) = \begin{cases} 0, & x < a; \\ \frac{x-a}{b-a}, & a \leq x \leq b; \\ 1, & x > b. \end{cases} \quad (1.5)$$

Calcul d'inverse : On a,

$$\mathbb{F}(x) = u, \text{ avec } u \in [0, 1] \Rightarrow \frac{x-a}{b-a} = u \Rightarrow x = u(b-a) + a.$$

Donc, pour simuler une variable aléatoire uniforme sur  $[a, b]$ , il suffit

de générer des nombres aléatoires  $u_i$  de variable aléatoire uniforme sur l'intervalle  $[0, 1]$  et déduire les réalisations  $x_i$ , tel que :  $x_i = a + (b - a)u_i$ .

**2. Méthode de rejet :** Le but est de simuler une variable aléatoire  $X$  de densité  $f(x)$  et de fonction de répartition  $\mathbb{F}(x)$ . La méthode du rejet évoque deux hypothèses sur la fonction  $f(x)$  :

- $f$  est définie sur un support compact, c'est-à-dire qu'elle prend la valeur nulle en dehors d'un certain intervalle fermé  $[a, b]$ . Cette méthode se résume comme suit :
  - a) Normaliser la fonction  $f(x)$  de telle façon que :  $f(x) = c \times g(x)$ , où  $c$  est une constante et  $g(x) \leq 1$ .
  - b) Générer une variable aléatoire  $U_1$  qui suit une loi uniforme sur  $[0, 1]$ .
  - c) Définir  $x$  comme fonction linéaire de  $u_1$ ,  $x = a + (b - a) \times u_1$ .
  - d) Générer une autre variable aléatoire  $U_2$  qui suit une loi uniforme sur  $[0, 1]$ . Si  $u_2 \leq g(x)$ , alors on accepte  $x$ , comme une réalisation de  $X$ . Sinon, on la rejette et on revient à l'étape **b**.
- Supposons qu'on dispose d'une fonction  $g(x)$  majorant de  $f(x)$ , C'est-à-dire :  $f(x) < g(x)$ ,  $\forall x$ . Si  $g(x)$  n'est pas une fonction de densité (c'est le cas en général), alors on prend :

$$c = \int_0^{\infty} g(t)dt, \quad \int_0^{\infty} f(t)dt = 1 \quad \text{et} \quad h(x) = \frac{g(x)}{c}.$$

Dans ce cas, les étapes précédentes b), c) et d) deviennent sous la forme suivante :

- a) Générer  $y$  selon  $h$ .
- b) Générer  $U \rightsquigarrow [0, 1]$ .



- c) Si  $u \leq \frac{f(y)}{g(y)}$ , alors on prend  $x = y$  ;  
 sinon, on rejette et on revient à l'étape **a**).

**3. Méthode de convolution :** La distribution de la somme de plusieurs variables aléatoires indépendantes est appelée : convolution des distributions initiales. Cette méthode consiste à sommer deux variables (ou plus) pour obtenir une variable aléatoire distribuée selon la loi de probabilité souhaitée (voir l'exemple 1.2.1). Si la variable aléatoire qu'on désire générer peut se représenté comme somme d'autres variables que l'on peut générer aisément :  $X = Y_1 + Y_2 + Y_3 + \dots + Y_n$ , alors on génère les  $Y_i$  et on obtient  $X$  en les sommant.

**Exemple 1.2.1**

- Une loi Gamma est une somme des lois Exponentielles de même paramètre.
- Une loi Khi-Deux ( $\chi_n^2$ ) à  $n$  degré de liberté est une somme de  $n$  lois Normales  $[0, 1]$  indépendantes au carré.
- Une loi Binomiale( $n, p$ ) est une somme de  $n$  lois Bernoulli( $p$ ).

**Génération d'une variable qui suit une loi normale** On dit que  $X$  suit une loi normale  $N(m, \sigma)$ , lorsque la densité de sa loi est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2}.$$

La fonction de répartition de la loi Normale ne peut pas être explicitée, donc la méthode d'inversion ne s'applique pas. On la génère de la manière suivante :

$$X = \sigma Y + m,$$

où  $Y$  est une variable aléatoire Normale centrée réduite, calculée à partir d'une suite de  $n$  variables aléatoires  $U_1, U_2, U_3, \dots, U_n$  uniformément distribuées entre  $[0, 1]$ .

$$Y = \frac{\sum_{i=1}^n U_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}}. \quad (1.6)$$

**Remarque 1.2.3** Cette écriture est justifiée par le théorème des grands nombres :  $\sum_1^n X_i \rightsquigarrow N(n\bar{X}, n\sigma_X), \forall$  la loi de  $X$ .

#### 4. Méthode de décomposition

Il s'agit de décomposer la fonction de répartition  $\mathbb{F}$  de la variable aléatoire  $X$  comme suit :

$$\mathbb{F}(x) = \sum_{i \geq 1} p_i \mathbb{F}_i(x). \quad (1.7)$$

La méthode se résume ainsi :

Écrire  $\mathbb{F}(x) = \sum_{i \geq 1} p_i \mathbb{F}_i(x)$  où :  $\sum_{i \geq 1} p_i = 1$ .

a) Générer  $U \rightsquigarrow U[0, 1]$ .

b) Trouver  $i$  tel que :  $0 + p_1 + p_2 + p_3 + \dots + p_{i-1} < U \leq p_1 + p_2 + p_3 + \dots + p_i$ .

c) Générer  $X$  selon  $\mathbb{F}_i(x)$ .

Pour s'assurer que les suites des nombres générées possèdent un comportement aléatoire selon la loi désirée, il faut tester ce comportement selon deux critères : l'Uniformité et l'Indépendance, tout en utilisant les tests d'Ajustement.

## 1.2.2 Quelques tests statistiques

Les tests d'ajustement ont pour but de vérifier si un échantillon provient ou pas d'une variable aléatoire de fonction de distribution connue  $\mathbb{F}_0(x)$ . Soit  $\mathbb{F}(x)$ , la fonction de répartition de la variable échantillonnée. Il s'agit de tester  $H_0$  "  $\mathbb{F}(x) = \mathbb{F}_0(x)$  " contre  $H_1$  "  $\mathbb{F}(x) \neq \mathbb{F}_0(x)$  " [1].

Les tests classiques les plus usuels sont :

### 1. Test de Khi-Deux [5]

Soit  $X_1, X_2, \dots, X_n$  un n-échantillon issu d'une variable aléatoire  $X$ . On partage le domaine  $D$  de la variable  $X$ , partie de l'ensemble des réels  $R$ , en  $r$  classes  $c_1, c_2, \dots, c_r$ . Généralement, on prend  $r \simeq \sqrt{n}$ .

Soit :

- $n_i$  : l'effectif de la classe  $c_i$ .
- $p_i$  : la probabilité de se trouver dans la classe  $c_i$ . Elle est déduite à partir de la loi de probabilité à tester.
- $n_i p_i$  : effectif théorique de la classe  $c_i$ .

Pearson a démontré que la variable aléatoire

$$K_n^2 = \sum_{i=1}^r \frac{(n_i - np_i)^2}{np_i}, \quad (1.8)$$

suit asymptotiquement une loi de Khi-Deux à  $(r - 1)$  degré de liberté.  $n_i$  étant une variable aléatoire représentant l'effectif de la classe  $c_i$  et dont la réalisation est  $n_i$ . Soit  $k_n^2$  la réalisation de la variable aléatoire  $K_n^2$ .

La règle de décision est alors sous la forme :

- Si  $k_n^2 < \chi_{(r-1, \alpha)}^2$ , on accepte l'ajustement de la variable aléatoire  $X$  par la loi choisie.

- Si  $k_n^2 \geq \chi_{(r-1, \alpha)}^2$ , on rejette l'ajustement de la variable aléatoire  $X$  par la loi choisie.

Lorsque les paramètres de la loi à valider sont estimés à partir de l'échantillon, le degré de liberté du Khi-Deux est alors égal à  $(r - q - 1)$ ,  $q$  étant le nombre de paramètres estimés. L'application du test Khi-Deux doit satisfaire les conditions suivantes :

- (a) Le nombre de classes doit être supérieur ou égal à 7.
- (b) L'effectif théorique de chaque classe doit être supérieur ou égal à 8.
- (c) Les effectifs théoriques des  $k$  classes doivent être sensiblement égaux.

## 2. Test de Kolmogorov-Smirnov [5]

Ce test est plus puissant que le précédent car, c'est celui dans lequel le risque d'accepter  $H_0$  à tort est plus faible. La procédure à suivre est la suivante :

- On tire un échantillon de  $n$  observations à l'aide du générateur ;
- On classe les observations selon un ordre croissant ;
- On compare la fonction de répartition empirique  $\mathbb{F}_n(x)$  calculée à partir de ces  $n$  nombres pseudo-aléatoires avec la fonction de répartition théorique  $\mathbb{F}(x)$  (loi uniforme sur  $[0, 1]$ ) et on calcul :

$$D_n = \max |\mathbb{F}_n(x) - \mathbb{F}(x)| = \max D(x_u), \quad (1.9)$$

où

$$\mathbb{F}_n(x) = \frac{\{\text{nombre d'observations} \leq x\}}{\text{la taille de l'échantillon}}$$

et

$$\mathbb{F}(x) = x, \quad \text{si } x \in [0, 1].$$

On fixe un seuil de signification  $\alpha$ , et soit  $d(\alpha)$  la valeur tabulée (obtenue

à partir de la table de Kolmogorov-Smirnov), tel que  $P(D_n > d(\alpha)) = \alpha$ .

La règle de décision est alors de la forme :

- On rejette  $H_0$  Si  $D_n > d(\alpha)$  ;
- On accepte  $H_0$  Sinon.

3. **Test de Student** Un des plus importants usages de la simulation est la comparaison des performances de deux ou plusieurs systèmes. Pour réaliser cette comparaison à partir d'échantillons (données statistiques) obtenus de la simulation des systèmes observés, nous devons recourir aux méthodes statistiques. Dans ce qui suit, nous présenterons l'une de ces méthodes à savoir le test de Student [5].

Etant donné deux échantillons de taille  $n_1, n_2$ , peut-on admettre qu'ils ont été prélevés dans une même population relativement à la variable étudiée.

Mathématiquement, le problème se formalise de la manière suivante :

On observe sur le premier échantillon (les réalisations d'une variable aléatoire  $X_1$ ), de fonction de répartition  $F_1(x)$  et sur le deuxième échantillon (les réalisations d'une variable aléatoire  $X_2$ ), de fonction de répartition  $F_2(x)$  ; on veut tester :

$$H_0 : F_1(x) = F_2(x) \quad \text{contre} \quad H_1 : F_1(x) \neq F_2(x).$$

Dans la pratique, on se contentera de vérifier l'égalité des espérances, donc on a à tester :

$$H_0 : \text{''}\bar{X} = m_{th}\text{''} \quad \text{contre} \quad H_1 : \text{''}\bar{X} \neq m_{th}\text{''},$$

où

$\bar{X}$  : est la moyenne de l'échantillon /  $m_{th}$  : est la moyenne théorique.

Ce test est basé sur la statistique suivante :

$$T_{n-1} = \frac{(\bar{X} - m_{th})\sqrt{n-1}}{S}, \quad (1.10)$$

avec :

$$S^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2.$$

$H_0$  est acceptée si :

$$T_{n-1} < t_{(n-1, \frac{\alpha}{2})}, \quad (1.11)$$

où  $t_{(n-1, \frac{\alpha}{2})}$  est la quantité de Student au seuil  $\alpha$ .

## Conclusion

La simulation à événements discret est un outil puissant et universel. Les gains à tirer d'une expérience de simulation sont variés : description (validation d'une architecture), explication (observation, expérimentation sur une maquette), et prédiction (mesure de performances, ou prévision de comportement). Son champ d'application recouvre les problèmes d'analyse, de conception, d'optimisation.

La simulation nous permet d'étudier les conditions d'opération extrême du système et d'évaluer les conséquences sans mettre en danger le système ou son environnement. Permet de concevoir des systèmes très complexes, elle devient le moyen le plus sûr pour la compréhension d'un système quand son étude s'avère difficile. C'est un instrument idéal de "décideur", aussi bien que de technicien.

Malgré tout ce qu'elle présente comme avantages la simulation est un outil coûteux en termes de temps de calcul, consommateur de ressources informatiques, et ne fournit que des estimations de ce que l'on cherche.

## Chapitre 2

# Systèmes d'attente Markoviens avec Bernoulli feedback

### Introduction

Dans ce chapitre, une brève introduction sur la théorie des files d'attente sera introduite dans un premier lieu. Par la suite, nous présenterons quelques systèmes de files d'attente markovien ( $M/M/1$ ,  $M/M/m$ ,  $M/M/\infty$ ) avec Bernoulli feedback de clients et certaines de leurs caractéristiques.

### 2.1 Définition des files d'attente

Une file d'attente (queue) est un système stochastique composé d'un certain nombre (fini ou non) de place d'attente d'un ou plusieurs serveurs et bien sûr de clients qui arrivent, attendent, se font servir selon des règles de priorités données, et quittent le système.

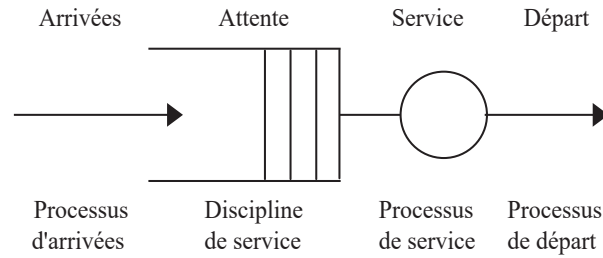


FIGURE 2.1 – Représentation d'une file d'attente simple.

## 2.2 Notation de Kendall-Lee

La notation de Kendall permet de décrire les files d'attente qui ont les propriétés ci-dessous :

► Les instants auxquels les clients arrivent est un processus de renouvellement en temps discret ou continu. Autrement dit, les durées entre chaque arrivée successive sont identiquement indépendamment distribuée (*i.i.d*), de loi  $A$  sur  $\mathbb{N}$  ou  $[0; \infty[$ . On numérote les différents clients à partir de 0 et l'on suppose que le client 0 arrive au temps 0. Ainsi, si  $T_n$  représente la durée entre l'arrivée du  $n^{ieme}$  client et celle du  $(n + 1)^{ieme}$  client, les  $T_n$  sont *i.i.d*, de loi  $A$  et les instants d'arrivée sont  $0, T_1, T_2, T_3, \dots$  [10].

► Les temps de service des différents clients sont *i.i.d*, et indépendants du processus des entrées. On note  $B$  la loi du temps de service, et  $U_n$  le temps de service du  $n^{ieme}$  client. Ainsi  $U_0, U_1, \dots$  sont *i.i.d*. de loi  $B$  et indépendants de  $T_n$ .

Un modèle est spécifié par une suite de six symboles :

$$A/S/m/K/P/D$$

La signification de chaque symbole est :

–  $A$  : nature du processus des arrivées,



- $S$  : nature du processus de service,
- $m$  : nombre de serveurs,
- $K$  : capacité d'accueil de la file d'attente,
- $P$  : taille de la population,
- $D$  : discipline de la file.

Dans la description des processus d'arrivée et de service, les symboles les plus courants sont :

- $M$  : loi Exponentielle (Memoryless).
- $E_k$  : loi d'Erlang- $k$ .
- $\Gamma$  : loi Gamma.
- $D$  : loi Déterministe (taux d'inter-arrivées ou de service constant).
- $G$  : loi Générale (quelconque).
- $GI$  : loi Générale Indépendante.
- $H_k$  : loi Hyper exponentielle- $k$ .
- $C_k$  : loi de Cox -  $k$ .
- $PH_k$  : loi de type "phase" à  $k$  étapes.

Lorsque les trois éléments de la notation de Kendall ne sont pas précisés, il est sous-entendu que  $K = \infty$ ,  $P = \infty$ ,  $D = FIFO$ .

**Processus d'arrivée :** Les clients arrivent au sein du système en décrivant un processus déterminé. Ils peuvent par exemple être régulières et leurs arrivées sont espacées par un temps égal soit à  $t$  (c'est-à-dire : chaque  $t$  unité de temps, on a une arrivée) mais ce cas est rare et assez difficile à schématiser mathématiquement.

Le modèle le plus simple et le plus courant est celui des arrivées complètement aléatoires, ce qui est caractérisé par le processus de Poisson [1].

**Processus de service :** La deuxième composante d'un système de files d'attente est la quantité de service demandée par un client. Dans la majorité des cas, on

suppose que la population des clients est homogène, ce qui entraîne que les services demandés sont identiquement distribués, ou ont une distribution commune dite distribution de service.

En pratique, on rencontre la distribution exponentielle qui est la plus simple à manipuler mathématiquement. Une propriété assez importante de cette distribution est son manque de mémoire, qui pourrait être caractérisé par le fait que le temps résiduel d'un service est indépendant du temps déjà écoulé de ce service.

**Nombre de serveurs :** Une station peut contenir plusieurs serveurs disposés en parallèle ou en série. Soit  $m$  le nombre de serveurs. Dès qu'un client arrive à la station, soit il trouve un serveur libre et le client entre immédiatement en service, soit tous les serveurs sont occupés et le client se place dans la file en attente d'un serveur qui se libère.

Pour définir une station multiserveurs, il faut définir la distribution de service de chacun des serveurs. La plupart du temps, les serveurs sont supposés identiques (ils possèdent donc la même distribution) et indépendants les uns des autres. Une station particulière est la station IS (Infinity Servers), dans laquelle le nombre de serveurs est infini. Cette station ne comporte donc pas de file d'attente. Dès qu'un client s'y présente, il trouve en effet instantanément un serveur disponible, et entre directement dans le service. Elle permet de modéliser des phénomènes de retard pur, le temps passé dans la station étant indépendant du nombre de clients simultanément présents et se réduit donc au temps de service.

**Capacité de la file :** La capacité de la file peut être finie ou infinie. Lorsqu'elle est limitée et qu'un client arrive alors que cette file est pleine, le client est perdu, ou rentre dans l'orbite, s'il s'agit des systèmes avec rappels.

**Discipline du service :** La discipline de service détermine l'ordre dans lequel

les clients sont rangés dans la file et y sont retirés pour recevoir un service. Les disciplines les plus courantes sont :

- ▶ FIFO (First In First Out) ou PAPS (Premier Arrivé Premier Servi) : C'est la file standard, dans laquelle les clients sont servis suivant l'ordre de leur arrivée.
- ▶ LIFO (Last In First Out) ou DAPS (Dernier Arrivé Premier Servi) : Cela correspond à une pile, dans laquelle le dernier client arrivé (donc posé sur la pile) sera le premier traité (retiré de la pile).
- ▶ RANDOM (Aléatoire) : Les clients accèdent au serveur de manière aléatoire, indépendamment de l'ordre des arrivés.
- ▶ Round-Robin (Cyclique) : Tous les clients de la file d'attente entrent en service à tour de rôle, effectuent un quantum  $Q$  de leur temps de service et sont remplacés dans la file, jusqu'à ce que leur service soit totalement accompli. Cette discipline de service a été introduite afin de modéliser des systèmes informatiques en particulier.
- ▶ PS (Processor Sharing) : C'est le cas limite de la distribution Round-Robin lorsque le quantum de temps  $Q$  tend vers 0. Tous les clients sont servis en même temps, mais avec une vitesse inversement proportionnelle au nombre de clients simultanément présents. Si le taux du service est égal à  $\mu$  et qu'un instant donné il y a  $n$  clients à la station, tous les clients sont donc servis simultanément avec un taux  $\frac{\mu}{n}$ .
- ▶ Avec priorité : Chaque client a une priorité (statique ou dynamique c'est-à-dire absolue ou relative), le serveur sélectionne le client de haute priorité.
  - **Priorité relative** : Un client accède au service selon sa priorité. La file est gérée par ordre de priorité de la plus forte à la plus faible.

- **Priorité absolue** : Le service d'un client est interrompu lorsqu'un client de priorité supérieure se présente devant la file d'attente. Le client dont ce service est interrompu est remis en tête de la file.

## 2.3 Mesures de performance d'une file d'attente

L'étude d'une file d'attente ou d'un réseau de files d'attente a pour but de calculer ou d'estimer les performances d'un système dans des conditions de fonctionnement données. Ce calcul se fait le plus souvent pour le régime stationnaire uniquement et les mesures les plus fréquemment utilisées sont :

- $L$  : Nombre moyen de clients dans le système,
- $Q$  : Nombre moyen de clients dans la file d'attente,
- $T$  : Temps moyen de séjour d'un client dans le système,
- $W$  : Temps moyen d'attente d'un client dans la file,
- $U$  : Taux d'utilisation de chaque serveur,

Ces mesures ne sont pas indépendantes les unes des autres, mais sont liées par les relations suivantes [4] :

- $L = \lambda T$  ( Formule de Little), où  $\lambda$  représente le taux d'arrivées.
- $Q = \lambda W$ .
- $T = W + \frac{1}{\mu}$ , où  $\mu$  représente le taux de service.
- $L = Q + \rho$ , tel que  $\rho$  est la charge du système.

De manière générale, une file d'attente est stable si et seulement si le nombre moyen d'arrivées de clients par unité de temps, noté  $\lambda$ , est inférieur au nombre moyen de clients pouvant être servis par unité de temps. Si chaque serveur peut

traiter  $\mu$  clients par unité de temps et si le nombre de serveurs est  $m$ , une file est stable si et seulement si :

$$\lambda < m\mu \Leftrightarrow \rho = \frac{\lambda}{m\mu} < 1$$

Plusieurs variantes existent pour la modélisation de files d'attente selon la nature et le comportement du système. Entre autres on va s'intéresser aux files d'attentes Markoviennes avec Feedback,

## 2.4 Files d'attente Markoviennes avec Feedback

Dans la théorie des files d'attente, une file d'attente classique peut être décrite comme un système dans lequel les clients arrivent selon un processus d'arrivées, pour être servis par une installation de service selon un processus de service. Cependant, en pratique différents comportements du (des) serveur(s) et des clients peuvent être identifiés. Dans ce document on s'intéresse à la situation où tous les clients demandent le service principal et quelques-uns ont besoin de demander un autre service supplémentaire, que l'on peut distinguer en pratique, elle est décrite par la notion de "feedback".

### 2.4.1 Système d'attente $M/M/1$ avec Bernoulli feedback

Considérons un système de file d'attente  $M/M/1$  avec Bernoulli feedback. Cette dernière peut modéliser un guichet unique où chaque client reçoit un service dont la durée est une variable exponentielle de paramètre  $\mu$ , et le processus d'arrivée des clients dans la file est un processus de Poisson de taux  $\lambda$  (le nombre de clients  $N(t)$  arrivant pendant un intervalle de temps  $[0, t]$  suit une distribution de Pois-

son). Après avoir obtenu un service, avec une probabilité  $\beta' = 1 - \beta$ ; le client peut rejoindre le système en tant que client Bernoulli feedback pour recevoir un autre service supplémentaire. Sinon, il quitte définitivement le système, avec une probabilité  $\beta$ ; (où  $\beta' + \beta = 1$ ).

Ce système peut être schématiser tel qu'il est présenté dans la figure 2.2.

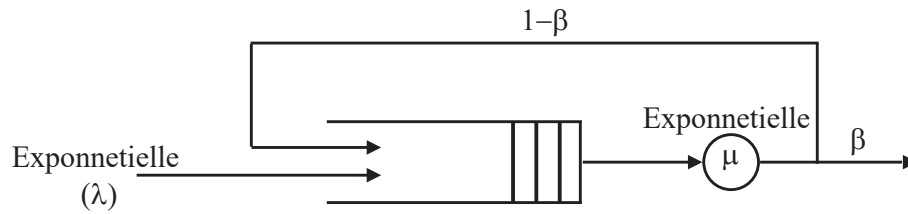


FIGURE 2.2 – Representation d'une file d'attente  $M/M/1$  avec Bernoulli feedback.

Notons que le processus des arrivées  $N(t)$  est décrit entièrement par :

$$P[N(t) = k] = e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad k = 0, 1, 2, \dots (\text{loi de Poisson de paramètre } \lambda t) \quad (2.1)$$

La durée des inter-arrivées  $T$  sont exponentiellement distribuées de fonction de densité :

$$a(t) = \lambda e^{-\lambda t}, \quad \forall t > 0.$$

Les durées de service  $S$  sont exponentiellement distribuées de fonction de densité :

$$b(t) = \mu e^{-\mu t}, \quad \forall t > 0.$$

Soit  $L(t)$  : le nombre de clients dans le système à l'instant  $t$ . Les probabilités d'état du système en régime transitoire sont données comme suit :

$$P_t = P[L(t) = n] \quad \text{et} \quad P_{ij}(dt) = P(L(t + dt) = j / L(t) = i),$$

avec

$$P_{ij}(dt) = \begin{cases} 1 - \lambda dt, & \text{si } i=j=0, \\ \lambda dt, & \text{si } j=i+1 \text{ et } j \geq 1, \\ \beta \mu dt, & \text{si } j=i-1 \text{ et } j \geq 0, \\ 1 - (\lambda + \beta \mu) dt, & \text{si } i=j \text{ et } j \geq 1, \\ 0, & \text{sinon.} \end{cases}$$

Le système peut être décrit par le système différentiel suivant (équation de Chapman-Kolmogorov) :

$$\begin{cases} P_0(t + dt) = P_0(t)(1 - \lambda dt) + P_1(t)\beta \mu dt, \\ P_n(t + dt) = P_{n-1}(t)\lambda dt + P_n(t)(1 - (\lambda + \beta \mu) dt) + P_{n+1}(t)\beta \mu dt, \quad n \geq 1. \end{cases}$$

Les probabilités d'état de système en question en régime stationnaire sont données comme suit :

$$\begin{cases} \lambda P_0 = \beta \mu P_1, & \text{si } n=0, \\ (\lambda + \beta \mu) P_n = \lambda P_{n-1} + \beta \mu P_{n+1}, & \text{si } n \geq 1. \end{cases}$$

- **Le taux d'utilisation du serveur** : par définition, le taux d'utilisation est la probabilité pour que le serveur de la file soit occupé :

$$U = \delta = \frac{\lambda}{\beta \mu}. \quad (2.2)$$

- **La probabilité qu'il y ait  $n$  clients dans le système à l'instant d'entrée** est :

$$\pi_n = \pi_0 \delta^n, \quad n = 0, 1, 2, \dots \text{ avec : } \pi_0 = 1 - \delta. \quad (2.3)$$

- **Le nombre moyen de clients dans le système et dans la file :**

$$L = \frac{\delta}{1 - \delta} \quad , \quad Q = \frac{\delta^2}{1 - \delta}. \quad (2.4)$$

- **Le temps moyen de séjour et d'attente :**

$$T = \frac{1}{\beta\mu - \lambda} \quad , \quad W = \frac{\lambda}{\beta\mu(\beta\mu - \lambda)}. \quad (2.5)$$

**Remarque 2.4.1** *La propriété "sans mémoire" de la loi exponentielle (décrivant un temps de service) veut dire que la probabilité pour que le service se termine avant le temps supplémentaire  $t_0$  sachant qu'il a débuté  $t_1$  instants de temps plus tôt, ne dépend pas du temps  $t_1$  que le client a déjà passé en service.*

## 2.4.2 Système d'attente $M/M/m$ avec Bernoulli feedback

On considère un système identique à la file donné dans (2.4.1) excepté qu'il contient  $m$  serveurs identiques et indépendants les uns des autres. On conserve les mêmes hypothèses : processus d'arrivée des clients est Poissonnien de taux  $\lambda$  et le temps de service exponentielle de taux  $\mu$  (pour chacun des serveurs). Après avoir obtenu un service, avec une probabilité  $\beta' = 1 - \beta$  ; le client peut rejoindre le système en tant que client Bernoulli feedback pour recevoir un autre service supplémentaire. Sinon, il quitte définitivement le système, avec une probabilité  $\beta$ . Ce système est connu sous le nom de la file  $M/M/m$  avec Bernoulli feedback [14] où sa condition de stabilité est donnée comme suit :

$\lambda < m\mu\beta$  (signifie que le nombre moyen de clients qui arrivent à la file par unité de temps doit être inférieur au nombre moyen de clients que les serveurs de la file en question sont capables de traiter par unité de temps).



Les probabilités d'état du système en régime transitoire sont données comme suit :

$$P_{ij} = \begin{cases} \lambda dt, & si \quad j=i+1, \\ i\beta\mu dt, & si \quad j=i-1 \text{ et } i \leq m, \\ m\beta\mu dt, & si \quad j=i-1 \text{ et } i \geq m, \\ 1 - (i\beta\mu + \lambda)dt, & si \quad i=j \text{ et } i \leq m, \\ 1 - (m\beta\mu + \lambda)dt, & si \quad j=i \text{ et } i \geq m, \\ 0, & sinon. \end{cases}$$

Le système peut être décrit par le système différentiel suivant (équation de Chapman-Kolmogorov) :

$$\begin{cases} P_0(t + dt) = P_0(t)(1 - \lambda dt) + P_1(t)\beta\mu dt, \\ P_n(t + dt) = P_{n-1}(t)\lambda dt + P_n(t)(1 - (\lambda + i\beta\mu)dt) + P_{n+1}(t)(i + 1)\beta\mu dt, \quad n \leq m, \\ P_n(t + dt) = P_{n-1}(t)\lambda dt + P_n(t)(1 - (\lambda + m\beta\mu)dt) + P_{n+1}(t)m\beta\mu dt, \quad n \geq m. \end{cases}$$

Les probabilités d'état de système en question système en régime stationnaire sont données comme suit :

$$\begin{cases} \lambda P_0 = \beta\mu P_1, & si \quad n=0, \\ (\lambda + i\beta\mu)P_n = \lambda P_{n-1} + (i + 1)\beta\mu P_{n+1}, & si \quad n \leq m, \\ (\lambda + m\beta\mu)P_n = \lambda P_{n-1} + m\beta\mu P_{n+1}, & si \quad n \geq m. \end{cases}$$

Les caractéristiques de ce système en régime stationnaire sont données par les relations suivantes :

- **La probabilité qu'il y ait n clients dans le système à l'instant d'entrée est :**

$$\pi_n = \begin{cases} \frac{(\frac{\lambda}{\beta\mu})^n}{n!} \pi_0, & si \quad n \leq m, \\ \frac{(\frac{\lambda}{\beta\mu})^n}{m!m^{n-m}} \pi_0, & si \quad n > m, \end{cases} \quad (2.6)$$

où :

$$\pi_0 = \left[ \sum_{n=0}^m \frac{\left(\frac{\lambda}{\beta\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\beta\mu}\right)^{m+1}}{m!(m - \frac{\lambda}{\mu})} \right]^{-1},$$

$$\zeta = P(\text{attente}) = P(X \geq m) = \frac{\pi_m}{1 - \delta}.$$

- Le taux d'utilisation de chaque serveur est :

$$U = \delta = \frac{\lambda}{m\beta\mu} \quad (2.7)$$

- Le nombre moyen de clients présents dans le système et en attente :

$$L = m\delta + \frac{\delta\zeta}{1 - \delta}, \quad Q = \frac{\delta\zeta}{1 - \delta}. \quad (2.8)$$

- Le temps moyen de séjour et d'attente sont :

$$T = \frac{1}{\beta\mu} \left(1 + \frac{\zeta}{m(1 - \delta)}\right), \quad \bar{W} = \frac{\zeta}{m\beta\mu(1 - \delta)}. \quad (2.9)$$

#### Remarque 2.4.2

Un cas particulier de ce système est bien que lorsque considère qu'il est composé d'un nombre illimité de serveurs identiques et indépendants les uns des autres. Ce système est connu sous le nom de  $M/M/\infty$  avec Bernoulli feedback. Ainsi, ses caractéristiques peuvent être quantifier facilement ou tout simplement deduite de celle du système précédent et on aura ce qui suit :

$P\{N = n\}$  : La loi stationnaire du nombre de clients dans le système pour tout entier  $n$  est

$$P\{N = n\} = \pi_n = \pi_0 \frac{\left(\frac{\lambda}{\beta\mu}\right)^n}{n!}, \quad n = 0, 1, \dots \quad (2.10)$$

$\pi_0$  : En sommant sur tous les entiers et en remarquant que cette somme doit être égale à l'unité des probabilités totale nous pouvons déduire,  $\pi_0$

$$\pi_0 = e^{-\left(\frac{\lambda}{\beta\mu}\right)}.$$

$L$  et  $Q$  : Le nombre moyen de clients dans le système et en attente sont

$$L = \frac{\lambda}{\beta\mu}, \quad Q = 0. \quad (2.11)$$

$T$  et  $W$  : Le temps moyen de séjour et d'attente sont :

$$T = \frac{1}{\beta\mu}, \quad W = 0$$

## Conclusion

L'une des hypothèses considérées dans ce chapitre, l'analyse exacte des systèmes d'attentes  $M/M/1$  et  $M/M/m$  avec Bernoulli feedback des clients, est que la probabilité de feedback est une constante. Cependant, en pratique on peut constater que la demande d'un service supplémentaire dépend du nombre de clients en attente autrement dit la probabilité de demander d'un service supplémentaire est décrite par une suite décroissante en fonction du nombre de clients dans la file. On parle alors, par abus de langage, de systèmes d'attentes avec Feedback dépendante de l'état du système. Il est à signaler que, de tels systèmes sont difficiles à analyser par les approches exactes voire impossible dans certaines situations.

Dans le chapitre suivants, nous allons faire recours à la simulation pour analyser le comportement des performances du système d'attente  $GI/GI/1$  avec feedback dépendant en fonction de ses paramètres de départ.

## Chapitre 3

# Evaluation de performance d'un système d'attente avec feedback

### Introduction

Dans les deux chapitres précédents, nous avons présenté un aperçu général sur la technique de simulation et quelques systèmes de files d'attente Markoviens avec Bernoulli feedback de clients ainsi que certaines de leurs caractéristiques au régime stationnaire.

Dans ce chapitre, nous nous intéressons à l'évaluation des performances de tels systèmes d'attente lorsqu'on s'éloigne de l'hypothèse du processus de Markov et/ou la probabilité de feedback dépend du nombre de clients dans le système. En effet, nous allons analyser un système de files d'attente ( $GI/GI/1$ ) avec feedback de clients dépendant de l'état du système dont les lois des inter-arrivées, et des durées de service sont générales.

Cependant, vue la complexité de ce type de systèmes même dans leurs forme la

plus simple ceci rend leurs analyse exacte très difficile. Ainsi, pour répondre à notre objectif, nous avons eu recours à la technique de la simulation à événements discrets.

### 3.1 Plan de simulation

Nous allons simuler le système de files d'attente de type  $(GI/GI/1)$  avec feedbacks de clients tout en considérons des cas particulier de ce système à savoir :  $M/GI/1$ ,  $GI/M/1$  et le système  $M/M/1$ . Plus précisément, nous allons considérer les sept systèmes suivants :

$S_1$  : Il s'agit du système de files d'attente  $M/M/1$ , dont les durées entre deux arrivées consécutives suivent une loi exponentielle de taux  $\lambda$ , et les durées de service suivent aussi une loi exponentielle mais de taux  $\mu$ .

$S_2$  : Il s'agit du système  $GI/M/1$ , dont les durées entre deux arrivées consécutives suivent une loi de Weibull de paramètres  $(\lambda_1, \lambda_2)$ , et les durées de service distribuées selon une loi exponentielle du taux  $\mu$ .

$S_3$  : Il correspond au système d'attente  $M/GI/1$ , dont les durées entre deux arrivées consécutives sont distribuées selon une la loi exponentielle de taux  $\lambda$ , et les durées de service suivent une loi de Weibull de paramètres  $(\mu_1, \mu_2)$ .

$S_4$  : Il correspond au système  $GI/GI/1$ , dont les durées entre deux arrivées consécutives suivent une loi de Weibull de paramètres  $(\lambda_1, \lambda_2)$ , et les durées de service suivent une loi de Weibull de paramètres  $(\mu_1, \mu_2)$ .

$S_5$  : Il s'agit du système de files d'attente  $GI/M/1$ , dont les durées entre deux arrivées consécutives suivent une loi d'Erlang d'ordre 2 et de paramètre  $\lambda$ , et les durées de service suivent aussi une loi exponentielle de taux  $\mu$ .

$S_6$  : Il s'agit du système  $M/GI/1$ , dont les durées entre deux arrivées consécutives

sont distribuées selon une la loi exponentielle de taux  $\lambda$ , et les durées de service distribuées selon une loi d'Erlang d'ordre 2 et de paramètre  $\mu$ .

$S_7$  : Il correspond au système d'attente  $GI/GI/1$ , dont les durées entre deux arrivées consécutives sont distribuées selon une loi d'Erlang d'ordre 2 et de paramètre  $\lambda$ , et les durées de service suivent une loi d'Erlang d'ordre 2 et de paramètre  $\mu$ .

De plus, nous avons supposé dans le cas des sept systèmes que les clients servis peut quitter définitivement le système avec une probabilité  $\beta_n$ , avec  $n$  est le nombre de clients dans le système à l'instant de la fin de son service, ou il peut demander un éventuelle autre service (feedback) avec une probabilité  $1 - \beta_n$ . Pour la forme de cette dernière, nous avons considéré quatre situations (voir Figure 3.1) :

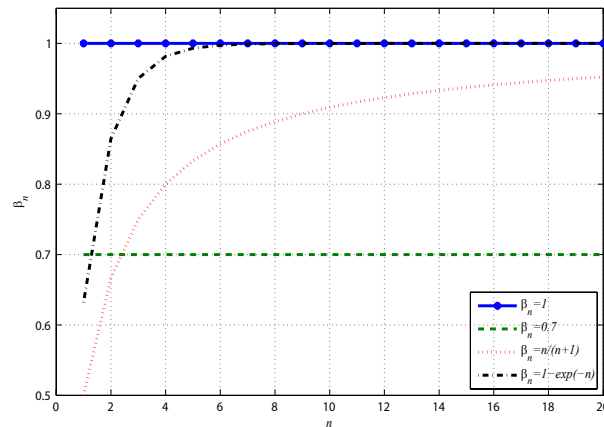


FIGURE 3.1 – File d'attente  $GI/GI/1$  avec feedback dépendant.

1.  $\beta_n = 1, \forall n \in \mathbb{N}^*$ , ce système est en réalité sans feedback de clients.
2.  $\beta_n = 0.7, \forall n \in \mathbb{N}^*$ , ce système est avec Bernoulli feedback qui ne dépend pas du nombre de clients dans le système.
3.  $\beta_n = \frac{n}{n+1}$ , pour  $n \in \mathbb{N}^*$ , ce système est avec un feedback de clients mais qui dépend du nombre de clients dans le système.

4.  $\beta_n = 1 - e^{-n}$ , pour  $n \in \mathbb{N}^*$ , ce système est avec un feedback de clients mais qui dépend exponentiellement du nombre de clients dans le système.

### 3.2 Simulation du système d'attente $GI/GI/1$

Dans cette section nous allons simuler le système  $GI/GI/1$  avec feedback de clients toute en respectant les étapes de simulation citées dans Section 1.1.2 du Chapitre 1. Enfin, le système considéré dans ce chapitre peut être présenté en sa forme réduite comme suit :

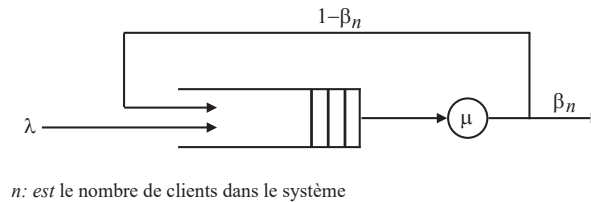


FIGURE 3.2 – File d'attente  $GI/GI/1$  avec feedback dépendant.

**Formulation du problème :** Comme on l'a signalé dans l'introduction du présent Chapitre, l'analyse analytique exacte d'un système d'attente de type  $GI/GI/1$  avec feedback dépendant s'avère très complexe et elle est inexistante dans la littérature. A cet effet, dans le but d'analyser ce type de systèmes nous allons exploiter l'approche simulation à événements discrets.

**Fixation des objectifs :** Supposons que notre objectif dans cette analyse et de fournir des estimations pour les mesures de performance du système en question et d'étudier l'influence de ses paramètres de départ sur ces performances. Plus précisément, l'objectif est de répondre aux questions suivantes :

1. Est-ce-que l'intensité du trafic influe sur les caractéristiques du système ?
2. Est-ce-que la probabilité de feedback et sa forme influe sur les caractéris-

tiques du système ?

3. Est-ce que la loi des inter-arrivées et celle des durées du service ont une influence sur les caractéristiques du système ?

**Elaboration du modèle :** Après une analyse minutieuse du comportement du système nous avons conclu que le programme, sous l'environnement MATLAB, du simulateur qui peut imiter ce comportement est le suivant (voir Figure 3.3) :

```

function simulationFeedBack(lambda,mu,beta_n,Tmax,ChoixA,ChoixS)
t=[0 inf inf];
n=0;
tsim=0;
while tsim<Tmax
    switch tsim
        case t(1)
            n=n+1;
            if n==1
                t(2)=t(1);
            end
            switch ChoixA
                case 1
                    ta=random('exp',1/lambda);           % Exponentielle
                case 2
                    ta=sum(random('exp',1/lambda,2,1)); % Erlang d'ordre 2
                case 3
                    ta=random('wbl',lambda(1),lambda(2));% Weibull
            end
            t(1)=t(1)+ ta;
        case t(2)
            switch ChoixS
                case 1
                    s=random('exp',1/mu);           % Exponentielle
                case 2
                    s=sum(random('exp',1/mu,2,1)); % Erlang d'ordre 2.
                case 3
                    s=random('wbl',mu(1),mu(2)); % Weibull
            end
            t(3)=t(2)+s;
            t(2)=inf;
        case t(3)
            u=random('unif',0,1);
            if u<beta_n(n)
                n=n-1;
            end
            if ne(n,0)
                t(2)=t(3);
            end
            t(3)=inf;
        end
        tsim=min(t);
    end
end

```

FIGURE 3.3 – Programme MATLAB du Simulateur de la file  $GI/GI/1$  avec feedback dépendant.

**Explication des étapes principales du programme :** Ici nous allons mettre



en évidence chaque étape de notre programme de simulation toute en faisons référence au comportement réel du système considéré.

**Etape 0.** Premièrement, on initialise les paramètres du système et du simulateur comme suit :

Fixer le système à simuler avec ces paramètre de départ (la loi d'inter-arrivées, la loi de service, la probabilité de feedback,...)

- $n = 0$  : Le système est initialement vide.
- $t_1 = 0$  : L'instant de la première arrivée au système.
- $t_2 = inf$  : L'instant du prochain début du service est fixé à une date ultérieure.
- $t_3 = inf$  : L'instant du prochain fin du service est fixé à une date ultérieure,
- $Tmin = min(t_1, t_2, t_3)$  : Cette étape consiste à initialiser l'horloge du simulateur à l'instant du premier événement. Dans notre cas, initialement  $Tmin = 0$ .

**Etape 1.** Si  $Tmin$  (l'horloge) a pas atteint le temps maximum de la simulation ( $Tmin > Tmax$ ) allez à l'**étape 3**; sinon ( $Tmin \leq Tmax$ ), on fait ce qui suit :

1. **Si**  $Tmin=t_1$  : Ceci signifie que l'événement qui dois ce produire est bien qu'une arrivée d'un nouveau client au système. **Alors** on doit procéder dans ce cas comme suit :
  - L'arrivée d'un nouveau client engendre l'augmentation du nombre de clients dans le système par une unité : ( $n = n + 1$ ).
  - **Si** le client est seul dans le système ( $n == 1$ ), **alors** il passe directement au serveur pour débiter son service  $t_2 = t_1$ , autrement

dit, l'instant du début de son service coïncide avec l'instant de son arrivée.

- **Sinon** le nouveau client n'est seul dans le système  $n \neq 1$ , **alors** il reste en attente dans la file.
- Le fait qu'on connaît la loi des inter-arrivées, on peut prévoir l'instant de la prochaine arrivée, il suffit de générer une variable aléatoire selon la loi des inters-arrivée ;  $a = \{ \text{générer une inter-arrivée} \}$ , qu'on additionne à l'instant de l'arrivée en cours :  $t_1 = t_1 + a$ .

2. **Si  $T_{\min}=t_2$**  : Cette étape signifie que l'événement actuelle est le début de service d'un client. Dans ce cas, les mises à jour qui doivent être effectuées sont :

- Le fait que le client commence son service à l'instant  $t_2$ , donc on peut prévoir l'instant de la fin de son service qui est :  $t_3 = t_2 + s$ , avec  $s$  est une variable aléatoire générée selon la loi des durées de service.
- La mise à jour du prochain début de service est initialisé à une date ultérieure  $t_2 = \text{inf}$ , car on ne peut pas savoir avec exactitude l'instant du prochain début de service.

3. **Si  $T_{\min}=t_3$**  : Dans ce cas, le client dans le serveur viens de terminer son service ; alors il peut quitter définitivement le système avec une probabilité  $\beta_n$  ou il rejoint à nouveau la queue de la file avec une probabilité complémentaire  $1 - \beta_n$  (feedback). Ainsi, on doit effectuer ce qui suit :

- Générer une probabilité uniforme  $u$  (selon une loi uniforme  $[0; 1]$ )
- Si  $\beta_n < u$  alors :
  - (a) Le client quitte le système par conséquent le nombre de clients sera diminué d'une unité ( $n = n - 1$ ).

- (b) **Si** il reste au moins un client en attente ( $n \neq 0$ ), **alors** y'aura un début de service du client en tête de la file, et ce sera au moment où le client à quitter le système c'est-à-dire  $t_2 = t_3$ .
  - (c) Par ailleurs, la mise à jour du prochain fin de service est initialisé à une date ultérieure  $t_3 = \inf$ , car on ne peut prévoir avec exactitude l'instant du prochain fin de service.
- Si  $\beta_n \geq u$  alors :
- (a) Le client ne quitte pas le système le fait qu'il a demandé un autre service par conséquent le nombre de clients reste inchangé dans le système et y a au moins un autres client dans le système. Alors, le début de service de celui en tête de la file sera immédiat ( $t_2 = t_3$ ).
  - (b) Par ailleurs, la mise à jour de la prochaine date du fin de service est initialisée à une date ultérieure  $t_3 = \inf$ , car on ne peut prévoir avec exactitude l'instant du prochain fin de service.

**Etape 2. Tmin = min( $t_1, t_2, t_3$ ) :** Cette étape consiste à faire incrémenter l'horloge du simulateur, en lui affectant l'instant d'un éventuelle future plus proche événement à exécuter et allez à l'**Etape 1**.

**Etape 3.** Afficher les résultats de simulation (les estimations des caractéristiques du système considéré).

**Etape 4.** Fin.

**Collecte des données :** Comme notre objectif est de simuler une file d'attente  $GI/GI/1$  avec feedback dépendant du nombre de clients dans le système alors, on a tendance à fixer les lois et les paramètres des inter-arrivées et du service, une intensité du trafic  $\rho \in \{1/15, 2/15, \dots, 10/15\}$  et  $\beta_n \in \{1, 0.7, \frac{n}{n+1}, 1 - e^{-n}\}$

avec  $n \in \mathbb{N}^*$ . Ceci tout en utilisant un temps simulation maximum qui assure la stationnarité du système ; dans notre application nous avons remarqué que les sept systèmes et les différentes valeurs de  $\rho$  considérées et de les quatre forme de  $\beta_n$  sont stationnaires pour un  $Tmax = 10000$ .

**Validation du modèle :** La validation du modèle se fera en exploitant le système d'attente  $M/M/1$  avec Bernoulli feedback comme témoin et ceci vue qu'on dispose des expressions exactes de ses mesures de performance. Pour cela, on fera appel au test de Student pour vérifier l'égalité entre les résultats analytiques (théoriques) et ceux obtenus par la simulation. C'est-à-dire de réaliser le test de conformité :

$$H_0 \text{ " } m = m_{th} \text{ " contre } H_1 \text{ " } m \neq m_{th} \text{ " .} \quad (3.1)$$

avec  $m$  est la valeur de la caractéristique fourni par le simulateur.

Dans ce cas, il suffit de calculer la réalisation de la statistique  $T$  donnée dans (1.10), et vérifier pour un seuil de risque  $\alpha$  l'inégalité (1.11), pour décider de rejeter ou non l'hypothèse  $H_0$ . Il est à noter que ceci est équivalent à vérifier si :

$$m_{th} \in [\bar{X}_{min}, \bar{X}_{max}] ,$$

où  $\bar{X}_{min}$  (resp.  $\bar{X}_{max}$ ) est la borne inférieure (resp. supérieure) de l'intervalle de confiance, à un seuil  $1 - \alpha$ , de la caractéristique  $X$  considérée.

### 3.3 Validation du modèle de simulation

Les caractéristiques théoriques du système  $S_1$ , pour les différents paramètres énumérés dans la section précédente pour  $\beta_n \in \{1, 0.7\}$  sont présentés dans la Table 3.1. Les résultats obtenus sur 100 simulations (échantillons) de ce même système

sont rangés dans la Table 3.2.

	$\rho$	0.0667	0.2000	0.2667	0.4000	0.4667	0.6000	0.6667
$\beta_n = 1$	$L$	0.0714	0.2500	0.3636	0.6667	0.8750	1.5000	2.0000
	$P_0$	0.9333	0.8000	0.7333	0.6000	0.5333	0.4000	0.3333
$\beta_n = 0.7$	$L$	0.1053	0.4000	0.6154	1.3333	2.0000	6.0000	20.000
	$P_0$	0.9048	0.7143	0.6190	0.4286	0.3333	0.1429	0.0476

TABLE 3.1: Performances théorique du système  $M/M/1$ .

D'après les résultats rangés dans les Tables 3.1 et 3.2, on constate que toutes les valeurs théoriques appartiennent aux intervalles de confiance, conçu à un seuil de confiance 95%, obtenues par la simulation. Á cet effet, l'hypothèse  $H_0$  du test 3.1 ne sera pas rejetée pour un risque de décision  $\alpha = 5\%$ . Ceci signifie que le modèle de simulation qu'on a conçu reproduit parfaitement le comportement d'un système d'attente  $GI/GI/1$  avec feedback de clients, d'où la validation du modèle. Alors, cela nous permet d'exploiter ce simulateur afin d'évaluer les performances du reste des systèmes décrits au début de la Section 3.1.

### 3.4 Effet de l'intensité de $\rho$ et de la probabilité

$$\beta_n$$

Notre objectif dans cette section est de répondre aux deux premières questions posées précédemment dans l'étape **Fixation des objectifs** dans la Section 3.2.

### 3.4.1 Résultats de simulation

L'exécution de notre simulateur pour le système  $S_1$  pour les différentes probabilités de feedback et des intensités du trafic exposées précédemment, nous a fourni les résultats rangés dans la Table 3.2 et ceux présentés dans les Figures 3.4–3.7.

	$\rho = \lambda/\mu$	$\beta_n = 1$			$\beta_n = 0.7$			$\beta_n = 1 - 1/(n + 1)$			$\beta_n = 1 - \exp(-n)$		
		$B_{inf}$	$Moy$	$B_{sup}$	$B_{inf}$	$Moy$	$B_{sup}$	$B_{inf}$	$Moy$	$B_{sup}$	$B_{inf}$	$Moy$	$B_{sup}$
$L(\beta_n)$	0.0667	0.0711	0.0713	0.0716	0.1047	0.1050	0.1053	0.1425	0.1429	0.1433	0.1106	0.1109	0.1113
	0.1333	0.1535	0.1539	0.1542	0.2351	0.2357	0.2364	0.3072	0.3079	0.3087	0.2342	0.2348	0.2353
	0.2000	0.2494	0.2500	0.2505	0.4000	0.4011	0.4023	0.4985	0.4995	0.5005	0.3725	0.3733	0.3740
	0.2667	0.3634	0.3642	0.3650	0.6137	0.6155	0.6172	0.7247	0.7263	0.7279	0.5304	0.5315	0.5325
	0.3333	0.4986	0.4996	0.5007	0.9064	0.9092	0.9121	0.9966	0.9987	1.0007	0.7127	0.7141	0.7155
	0.4000	0.6645	0.6659	0.6673	1.3315	1.3360	1.3405	1.3311	1.3338	1.3364	0.9275	0.9291	0.9308
	0.4667	0.8746	0.8766	0.8785	1.9895	1.9979	2.0063	1.7497	1.7536	1.7575	1.1856	1.1881	1.1905
	0.5333	1.1397	1.1427	1.1456	3.1762	3.1963	3.2163	2.2778	2.2831	2.2883	1.5023	1.5060	1.5096
	0.6000	1.4985	1.5033	1.5081	5.9326	5.9867	6.0408	2.9931	3.0016	3.0101	1.9181	1.9221	1.9262
0.6667	1.9967	2.0037	2.0108	19.6267	20.1378	20.6489	3.9789	3.9924	4.0060	2.4747	2.4816	2.4885	
$P_0(\beta_n)$	0.0667	0.9332	0.9334	0.9336	0.9048	0.9050	0.9052	0.8708	0.8711	0.8715	0.8973	0.8976	0.8978
	0.1333	0.8664	0.8666	0.8669	0.8089	0.8093	0.8097	0.7504	0.7509	0.7513	0.8001	0.8005	0.8009
	0.2000	0.7997	0.8000	0.8003	0.7132	0.7137	0.7142	0.6397	0.6401	0.6406	0.7093	0.7098	0.7103
	0.2667	0.7328	0.7331	0.7335	0.6184	0.6190	0.6195	0.5373	0.5380	0.5386	0.6236	0.6240	0.6245
	0.3333	0.6664	0.6668	0.6671	0.5234	0.5241	0.5248	0.4448	0.4454	0.4459	0.5430	0.5436	0.5441
	0.4000	0.5997	0.6001	0.6006	0.4275	0.4281	0.4288	0.3594	0.3600	0.3605	0.4682	0.4687	0.4693
	0.4667	0.5327	0.5331	0.5336	0.3333	0.3340	0.3347	0.2836	0.2841	0.2846	0.3982	0.3987	0.3991
	0.5333	0.4661	0.4666	0.4672	0.2374	0.2382	0.2391	0.2173	0.2178	0.2183	0.3332	0.3337	0.3342
	0.6000	0.3990	0.3997	0.4003	0.1423	0.1431	0.1440	0.1594	0.1600	0.1605	0.2724	0.2728	0.2733
0.6667	0.3320	0.3326	0.3332	0.0471	0.0480	0.0488	0.1109	0.1113	0.1118	0.2165	0.2170	0.2175	
$\bar{\beta}_n$	0.0667				0.6997	0.7005	0.7012	0.5165	0.5173	0.5180	0.6498	0.6507	0.6515
	0.1333				0.6993	0.6998	0.7003	0.5346	0.5352	0.5357	0.6685	0.6691	0.6696
	0.2000				0.6992	0.6996	0.7000	0.5552	0.5555	0.5559	0.6888	0.6892	0.6897
	0.2667				0.6996	0.7000	0.7003	0.5767	0.5771	0.5775	0.7091	0.7095	0.7099
	0.3333	1	1	1	0.6997	0.7000	0.7004	0.5999	0.6002	0.6005	0.7305	0.7309	0.7312
	0.4000				0.6995	0.6999	0.7002	0.6252	0.6255	0.6258	0.7529	0.7532	0.7535
	0.4667				0.6999	0.7002	0.7005	0.6516	0.6519	0.6522	0.7761	0.7764	0.7767
	0.5333				0.6997	0.7000	0.7003	0.6815	0.6818	0.6821	0.7997	0.8000	0.8003
	0.6000				0.6999	0.7001	0.7004	0.7142	0.7145	0.7149	0.8250	0.8253	0.8255
0.6667				0.6998	0.7000	0.7002	0.7497	0.7500	0.7504	0.8511	0.8514	0.8517	
$N(\beta_n)$	0.0667				1.4262	1.4277	1.4292	1.9304	1.9332	1.9360	1.5350	1.5370	1.5389
	0.1333				1.4279	1.4289	1.4299	1.8667	1.8686	1.8704	1.4934	1.4946	1.4958
	0.2000				1.4287	1.4295	1.4303	1.7989	1.8001	1.8013	1.4499	1.4509	1.4518
	0.2667				1.4279	1.4286	1.4294	1.7317	1.7329	1.7340	1.4086	1.4094	1.4101
	0.3333	1	1	1	1.4278	1.4285	1.4292	1.6653	1.6661	1.6670	1.3676	1.3683	1.3689
	0.4000				1.4281	1.4288	1.4295	1.5979	1.5986	1.5994	1.3271	1.3277	1.3282
	0.4667				1.4276	1.4281	1.4287	1.5332	1.5339	1.5346	1.2874	1.2880	1.2885
	0.5333				1.4280	1.4286	1.4291	1.4660	1.4666	1.4673	1.2496	1.2500	1.2505
	0.6000				1.4277	1.4283	1.4288	1.3989	1.3995	1.4001	1.2113	1.2117	1.2121
0.6667				1.4278	1.4283	1.4288	1.3326	1.3332	1.3339	1.1741	1.1745	1.1749	

TABLE 3.2: Résultats de simulation du système  $M/M/1$  avec feedback dépendant.

**Remarque 3.4.1** La lecture de la triplette  $(B_{inf} Moy B_{sup})$  se fait comme suit :

$B_{inf}$  : est la borne inférieure de l'intervalle de confiance à un seuil 95% de la caractéristique considérée.

$Moy$  : est la moyen empirique de la caractéristique considérée.

$B_{sup}$  : est la borne supérieur de l'intervalle de confiance à un seuil 95% de la caractéristique considérée.

**Exemple** : la triplette  $(0.0711 \ 0.0713 \ 0.0716)$  représente l'estimateur par intervalle de confiance et ponctuelle de la caractéristique de  $L$  (le nombre moyen de clients dans le système), où :

**0.0711** : est la borne inférieure de l'intervalle de confiance de  $L$ .

**0.0713** : est la moyen empirique (simulé) de  $L$  .

**0.0716** : est la borne supérieur de l'intervalle de confiance de  $L$ .

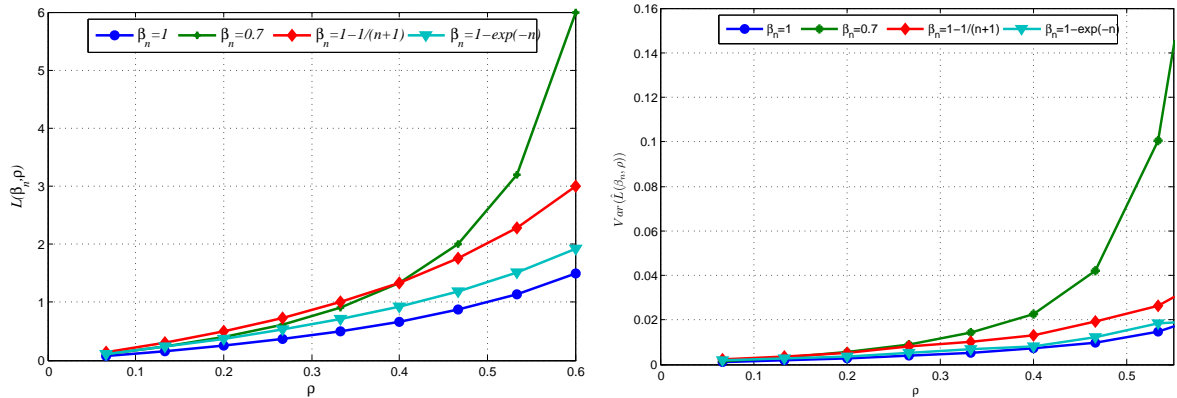


FIGURE 3.4 – Variation de  $L$  et  $Var(\hat{L})$  en fonction du couple  $(\beta_n, \rho)$ .

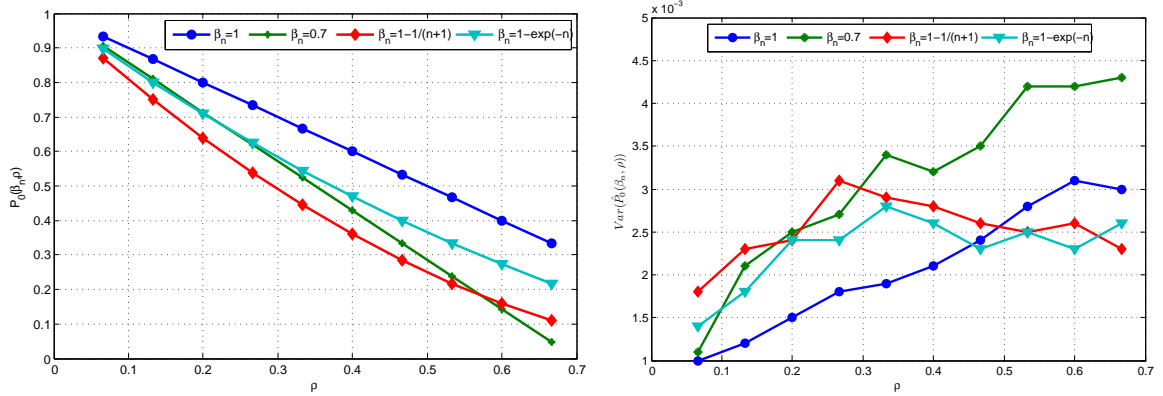


FIGURE 3.5 – Variation de  $P_0$  et  $Var(\hat{P}_0)$  en fonction du couple  $(\beta_n, \rho)$ .

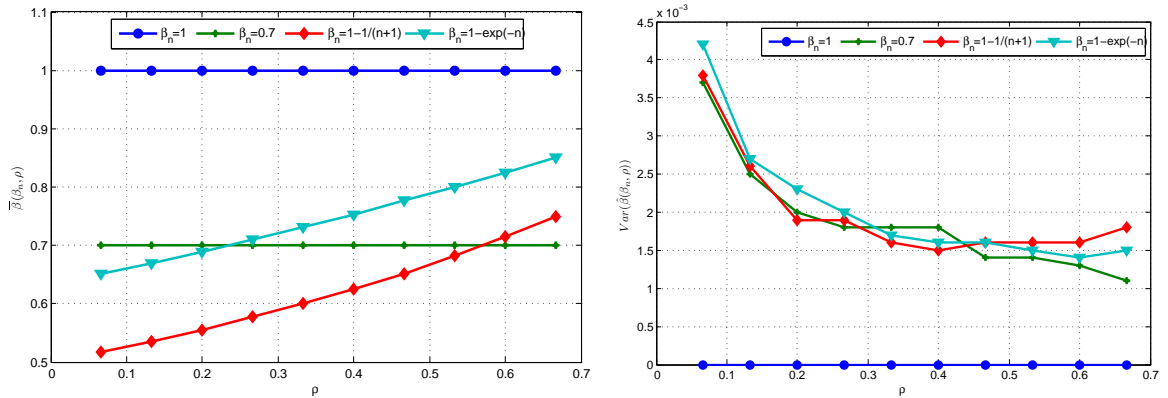


FIGURE 3.6 – Variation de  $\bar{\beta}$  et  $Var(\hat{\beta})$  en fonction du couple  $(\beta_n, \rho)$ .

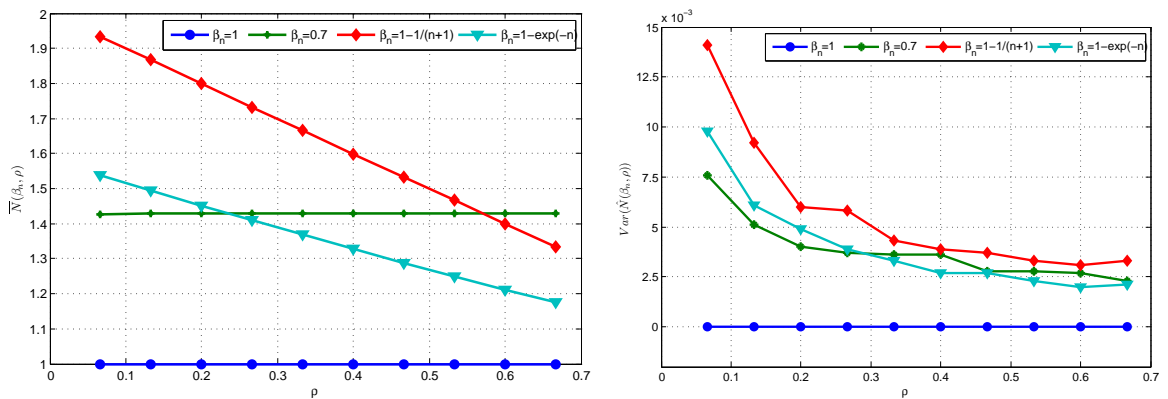


FIGURE 3.7 – Variation de  $N$  et  $Var(\hat{N})$  en fonction du couple  $(\beta_n, \rho)$ .



### 3.4.2 Analyse et interprétation des résultats

La réponse aux deux premières questions posées précédemment dans l'étape **Fixation des objectifs** (voir Section 3.2) sera obtenu après l'analyse et l'interprétation des résultats rangés dans la Table 3.2 et ceux présentés dans les Figures 3.4–3.7.

**L'effet de  $\beta_n$  et  $\rho$  sur  $L$  :** D'après les résultats numériques présentés dans les lignes associées au  $L(\beta_n, \rho)$  dans la Table 3.2 et les résultats graphiques présentés dans la Figure 3.4, on remarque que la croissance de l'intensité du trafic  $\rho$  engendre une nette augmentation du nombre moyen de clients dans le système  $L$ . De plus, la croissance de  $L$  en fonction de  $\rho$  dépend de la forme de la probabilité de feedback  $\beta_n$ . En effet, on constate que  $L$  est plus considérable dans le cas de  $\beta_n = 0.7$ , suivi respectivement par le cas où  $\beta_n = n/(n+1)$  et  $\beta_n = 1 - e^{-n}$  pour que  $L$  soit plus faible pour le cas où  $\beta_n = 1$  (sans feedback). Le constat fait sur le comportement de  $L$  en fonction de  $\rho$  et de  $\beta_n$  reste vraie pour  $Var(\hat{L})$ . Par conséquent, l'augmentation de la longueur de  $IC_{95\%}(L) = \left[ \bar{L} \pm 0.196 \sqrt{Var(\hat{L})} \right]$ .

Il est à noter que le comportement de  $L$  et  $Var(\hat{L})$  en fonction de  $\rho$  coïncide parfaitement avec les notions de la théorie des files d'attente où il est indiqué que le processus du nombre de clients dans le système tend à devenir un processus non stationnaire lorsque la charge du système se rapproche de 1 voir non stationnaire lorsque la charge est supérieure à 1 ( $L = \infty$  et  $Var(L) = \infty$ ).

**L'effet de  $\beta_n$  et  $\rho$  sur  $P_0$  :** D'après les résultats numériques présentés dans lignes associées à  $P_0(\beta_n, \rho)$  de la Table 3.2 et ceux présenté dans la Figure 3.5 on constate que l'augmentation de  $\rho$  engendre la décroissance de  $P_0$  ceci peut

s'expliquer par le fait que pour une grande charge du système ce dernier a tendance à être en permanence actif (entraîné de servir) c'est-à-dire  $U$  tend vers 1 par conséquent  $P_0$  tend vers 0 ( $U = 1 - P_0$ ). Tandis que le comportement de  $P_0$  en fonction de  $\beta_n$  il est exactement l'inverse de celui de  $L$ . La  $Var(\hat{P}_0)$  a globalement une tendance à être une fonction croissante en fonction de  $\rho$  mais pas d'une manière régulière. Ceci rend la longueur de l' $IC_{95\%}(\hat{P}_0)$  dans un certain sens aléatoire.

**L'effet de  $\beta_n$  et  $\rho$  sur  $\bar{\beta}$  :** D'après les résultats numériques présentés dans lignes associées à  $\bar{\beta}(\beta_n, \rho)$  de la Table 3.2 et ceux présentés dans la Figure 3.6, on constate qu'en fait et à mesure que  $\rho$  augmente  $\bar{\beta}$  tend vers 1 dans le cas  $\beta_n = n/(n+1)$  et  $\beta_n = 1 - e^{-n}$ , ceci peut s'expliquer par le fait que l'augmentation de  $\rho$  engendre l'augmentation de  $L$  (voir la discussion des résultats  $L(\rho, \beta_n)$ ) par conséquent les clients servis ont tendance à quitter le système plutôt que de demander un service supplémentaire car y a un nombre considérable de clients dans le système à la fin de son service ( $\beta_n$  est croissante en fonction de  $n$ ). On remarque dans ce cas que la  $Var(\bar{\beta})$  est pratiquement indépendante de la forme de  $\beta_n$  par conséquent la longueur de l' $IC_{95\%}(\bar{\beta})$  est pratiquement une constante lorsque on change  $\beta_n$ .

**L'effet de  $\beta_n$  et  $\rho$  sur  $N$  :** D'après les résultats numériques présentés dans lignes associées à  $N(\beta_n, \rho)$  de la Table 3.2 et ceux présentés dans la Figure 3.7, on remarque d'une part que pour une probabilité de feedback constante ( $\beta_n = 1$  et  $\beta_n = 0.7$ ) le nombre de service reçu par le même client est toujours le même ce qui est tout à fait naturelle. Tandis que pour le cas de  $\beta_n = n/(n+1)$  et  $\beta_n = 1 - e^{-n}$ , en fait et à mesure que  $\rho$  augmente le système a tendance à se comporter comme état un système sans feedback ( $N(\rho)$  tend vers 1

lorsque  $\rho$  tend vers 1). Ce dernier, résulte du fait

$$\lim_{\rho \rightarrow 1} L(\rho) = \infty \Rightarrow \lim_{L \rightarrow \infty} \beta_L = 1.$$

D'autre part, la  $Var(N)$  se rapproche de 0 en fur et à mesure que  $\rho$  augmente ce qui se traduit par la convergence de  $N$  vers une constante lorsque  $\rho$  tend vers 1. Ainsi, la longueur de l' $IC_{95\%}(N)$  est décroissante en fonction de  $\rho$  est tend à être nulle lorsque ce dernier tend vers 1.

En conclusion, les résultats obtenus dans cette application indiquent que les différentes caractéristiques considérées ( $L, P_0, \bar{\beta}$  et  $N$ ) dépend étroitement de la valeur de  $\rho$  et de la forme de  $\beta_n$ . De plus, ils nous fournis un constat de grande importance qui peut être résumé comme suit : Si pour deux forme de probabilités  $\beta_n^{(1)}$  et  $\beta_n^{(2)}$  de feedback on a  $\beta_n^{(1)} \leq \beta_n^{(2)}, \forall n$ , alors

$$L(\beta_n = 1, \rho) \leq L(\beta_n^{(2)}, \rho) \leq L(\beta_n^{(1)}, \rho).$$

### 3.5 Effet des distributions des processus d'arrivées et de service

Dans la présente section, notre objectif est de répondre à la troisième question fixée précédemment dans la section 3.2, à savoir la question concernant l'analyse de l'effet de la distribution du processus des arrivées et celle du processus de service sur les caractéristiques du système  $GI/GI/1$  avec feedback dépendant.

### 3.5.1 Résultats de simulation

L'exécution de notre simulateur pour les différents systèmes considérés précédemment ( $S_1 - S_7$ ) pour une durée de simulation maximale 10000 unité de temps nous a fournis les moyennes empiriques, des quatre caractéristiques :  $L$ ,  $P_0$ ,  $\bar{\beta}$  et  $N$ , rangées dans la Table 3.3 suivante.

$\beta_n$	Caractéristique	$\rho$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	
$\beta_n = 0.7$	$L(\beta_n, \rho, S_i)$	0.2	0.4014	0.3233	0.3626	0.3015	0.3420	0.3802	0.3275	
		0.4	1.3389	0.9522	1.0786	0.7420	1.0631	1.1930	0.9406	
		0.6	5.9491	3.9203	4.2682	2.3116	4.5389	5.0988	3.6591	
	$P_0(\beta_n, \rho, S_i)$	0.2	0.7136	0.7143	0.7141	0.7144	0.7145	0.7141	0.7139	
		0.4	0.4281	0.4286	0.4295	0.4290	0.4286	0.4298	0.4286	
		0.6	0.1441	0.1438	0.1438	0.1430	0.1434	0.1430	0.1439	
	$\bar{\beta}(\beta_n, \rho, S_i)$	0.2	0.7002	0.6997	0.6995	0.7000	0.6999	0.6997	0.7003	
		0.4	0.6999	0.7002	0.7003	0.7003	0.6997	0.7000	0.7002	
		0.6	0.7002	0.7003	0.7003	0.7003	0.7001	0.7002	0.7000	
	$N(\beta_n, \rho, S_i)$	0.2	1.4281	1.4292	1.4296	1.4287	1.4287	1.4291	1.4281	
		0.4	1.4288	1.4281	1.4280	1.4280	1.4292	1.4285	1.4281	
		0.6	1.4280	1.4279	1.4279	1.4279	1.4282	1.4280	1.4284	
	$\beta_n = 1 - 1/(n + 1)$	$L(\beta_n, \rho, S_i)$	0.2	0.5004	0.4440	0.4570	0.4171	0.4600	0.4788	0.4419
			0.4	1.3322	1.1328	1.1313	0.9500	1.1917	1.2258	1.0896
			0.6	2.9953	2.4916	2.3949	1.8822	2.6442	2.6796	2.3319
$P_0(\beta_n, \rho, S_i)$		0.2	0.6402	0.6199	0.6393	0.6168	0.6255	0.6388	0.6244	
		0.4	0.3601	0.3212	0.3544	0.3068	0.3350	0.3571	0.3297	
		0.6	0.1596	0.1232	0.1485	0.0988	0.1366	0.1549	0.1275	
$\bar{\beta}(\beta_n, \rho, S_i)$		0.2	0.5555	0.5265	0.5546	0.5219	0.5347	0.5544	0.5327	
		0.4	0.6246	0.5896	0.6197	0.5769	0.6017	0.6220	0.5969	
		0.6	0.7142	0.6842	0.7045	0.6657	0.6943	0.7094	0.6879	
$N(\beta_n, \rho, S_i)$		0.2	1.8001	1.8993	1.8031	1.9163	1.8701	1.8037	1.8772	
		0.4	1.6009	1.6961	1.6138	1.7334	1.6620	1.6076	1.6753	
		0.6	1.4000	1.4614	1.4193	1.5022	1.4402	1.4096	1.4535	
$\beta_n = 1 - \exp(-n)$		$L(\beta_n, \rho, S_i)$	0.2	0.3738	0.3370	0.3398	0.3203	0.3459	0.3550	0.3332
			0.4	0.9282	0.7898	0.7699	0.6697	0.8295	0.8469	0.7540
			0.6	1.9248	1.5391	1.4385	1.1167	1.6506	1.6623	1.4112
	$P_0(\beta_n, \rho, S_i)$	0.2	0.7095	0.6950	0.7092	0.6922	0.6994	0.7098	0.6982	
		0.4	0.4689	0.4366	0.4659	0.4248	0.4476	0.4670	0.4442	
		0.6	0.2726	0.2368	0.2667	0.2147	0.2501	0.2704	0.2434	
	$\bar{\beta}(\beta_n, \rho, S_i)$	0.2	0.6888	0.6552	0.6880	0.6496	0.6653	0.6884	0.6625	
		0.4	0.7526	0.7097	0.7492	0.6956	0.7242	0.7514	0.7192	
		0.6	0.8252	0.7867	0.8182	0.7637	0.8002	0.8221	0.7932	
	$N(\beta_n, \rho, S_i)$	0.2	1.4518	1.5264	1.4536	1.5395	1.5032	1.4526	1.5095	
		0.4	1.3287	1.4090	1.3347	1.4377	1.3808	1.3309	1.3904	
		0.6	1.2117	1.2711	1.2222	1.3094	1.2497	1.2164	1.2607	

TABLE 3.3: Résultats de simulation du système  $GI/GI/1$  avec feedback dépendant.

### 3.5.2 Analyse et interprétation des résultats

A partir des résultats rangés dans la Table 3.3 on constate dans l'ensemble les caractéristiques des sept systèmes  $S_1 - S_7$  dépendent étroitement des paramètres de départ définissant ce système à savoir : de l'intensité du trafic  $\rho$ , la distribution des durées des inter-arrivées, la distribution des durées de service ainsi que la forme de la probabilité  $\beta - n$ . En effet, on constate que

- La forme de  $\beta_n$  influe sur la valeurs des caractéristiques considérées, et ceci de la même manière que dans la première application où pour des valeurs moyennes et fortes de l'intensité du trafic on a :

$$L(\beta_n = 1, \rho) \leq L(\beta_n = 1 - e^{-n}, \rho) \leq L(\beta_n = n/(n + 1), \rho) \leq L(\beta_n = 0.7, \rho).$$

- La distribution des durées d'inter-arrivées et celles des durées de service influent sur les quatre caractéristiques considérées dans l'étude. De plus, leurs influences sont plus apparentes lorsque la charge du système est grande. Par contre, pour de faible charge du système nos résultats nous laisse croire que les caractéristiques des 7 systèmes sont pratiquement les même. Par conséquent, l'évaluation (l'approximation) des caractéristiques du système  $GI/GI/1$  avec feedback, dont l'analyse exacte est inexistante, via le système  $M/M/1$  est désormais justifié lorsque la charge du système est faible.
- Globalement, quand la charge de système augmente, le système  $GI/GI/1$  se comporte comme étant un système  $GI/GI/1$  sans feedback, ceci est le fait

que :

$$\lim_{\rho \rightarrow 1} L(\beta_n, \rho) = \infty \Rightarrow \begin{cases} N \text{ et } \bar{B} \text{ tend vers } 1, \\ Var(\hat{N}) \text{ et } Var(\hat{\beta}) \text{ tend vers } 0. \end{cases}$$

## Conclusion

Dans ce chapitre, l'analyse du modèle retenu par l'approche simulation à événements discrets, nous a donné un aperçu sur l'influence de l'intensité du trafic, les distributions des processus des inter-arrivées et du service, ainsi que la probabilité de feedback et sa forme sur quelques mesures de performance du système considéré dans l'étude.

## Conclusion générale

Le domaine de la pratique a pu exposer aux regards des spécialistes en mathématiques le chemin du développement. Dans ce mémoire, nous prouvons l'intérêt, l'utilité et l'application de la simulation dans les systèmes d'attente :  $M/M/1$ ,  $M/GI/1$ ,  $GI/M/1$  sans feedback et avec Bernoulli feedback, mais beaucoup plus aux systèmes complexes tels que  $GI/GI/1$  avec feedback dépendant qui a été l'objet de notre étude. A cet effet, on s'est intéressé à l'estimation des caractéristiques de ce système à savoir : le nombre moyen de clients dans le système  $L$ , la probabilité que le système soit vide  $P_0$  ( $1-P_0$  est la charge du système), la probabilité moyenne du feedback d'un client  $E(\beta_n) = \bar{\beta}$  et le nombre moyen de services reçu par un même client  $\bar{N}$ . La variation des estimations de ces dernières caractéristiques a été considérée également.

Après l'élaboration d'un modèle de simulation à événements discrets pour le système  $GI/GI/1$  avec feedback dépendant, sa validation a été faite par rapport aux systèmes Markoviens  $M/M/1$  sans feedback et  $M/M/1$  avec Bernoulli feedback, pour lequel des résultats théoriques existent. Le simulateur a été exploité pour mesurer les différentes performances du système  $GI/GI/1$  pour différents taux de charges variant entre 1/15 et 10/15 et pour des inter-arrivées et des services qui suivent une loi de Weibull de paramètres  $(\mu_1, \mu_2)$  ou une loi d'Erlang d'ordre deux de paramètre  $\mu$ .

Dans cette étude, l'analyse du modèle retenu par l'approche simulation, nous a donné la possibilité de réaliser une comparaison entre les résultats obtenus par cette dernière et ceux de l'analyse théorique lorsque ces derniers existent. De plus, les résultats obtenus nous ont permis d'identifier clairement la manière dont que la forme de la probabilité de feedback, l'intensité du trafic, ainsi que les lois des processus des inter-arrivées et du service influent sur les caractéristiques du système.

**Il sera intéressant de compléter cette étude :**

- **En prenant en considération, d'autres lois des inter-arrivées ou du service tel que : loi de Gamma, Cox, etc.**
- **Envisager d'autre cas particuliers sur la discipline de la file (FIFO, LIFO, Random, etc.)**
- **Envisager d'autres cas particuliers de la probabilité de feedback pour lesquels une analyse exacte est possible.**



# Bibliographie

- [1] Babes, M. (1995) Statistiques, Files d'attente et Simulation.
- [2] Banks, J. (1998) Handbook of simulation : Principles, methodology, advances, applications and practice. Wiley.
- [3] Baynat, B. (2000) Théorie des Files d'Attente. Des chaînes de Markov aux réseaux à forme produit. Hermès-Lavoisier.
- [4] Bose, S.K. (2002) An introduction to queueing systems. Kluwer Academic/ Plenum Publishers, 223 Spring Street, New York 10013 (USA)
- [5] Cherfaoui, M. (2016/2017) Polycopié de cours : Statistiques Appliquées à l'Expérimentation En Sciences Biologique, 3 année licence biologie. Département SNV, Université de Biskra.
- [6] Erard, P.J. et Déguénon, P. (1996) Simulation par événements discrets. Presses Polytechniques et Universitaires Romande.
- [7] Hêche, J.F. Liebling T.M. and Werra, D. (2003) Recherche Opérationnelle pour Ingénieurs II, Presses Polytechniques et Universitaires Romandes.
- [8] Perros, H. (2003) Computer Simulation Techniques : The Definitive Introduction. Computer Science Department NC State University Raleigh, NC.
- [9] Poisson, S.D. (1837) Recherches sur la probabilité des jugements en matière criminelle et en matière civile. Paris.

- [10] Pudlo, P. Cours de Master 1ère année : File d'Attente et Réseaux.
- [11] Sheldon, M.R. (2001) Simulation, Academic Press.
- [12] Sinclair, J.B. (2004) Simulation of computer systems and computer networks : A Process-Oriented Approach.
- [13] Toutain, L. (1991) Un simulateur pour systèmes répartis et temps-réel. PhD thesis, Université du Havre.
- [14] Werra, D.D. Liebling, T.M. et Heche, J.F. (2003) Recherche Opérationnelle pour ingénieurs : Tome 2. Presses Polytechniques et Universitaires Romandes.

## Annexe A.

# Simulation de lois usuelles et la fonction *RANDOM* sous Matlab

### 3.6 Simulation de lois usuelles

Dans cette section nous allons présenter quelques informations sur les loi utiliser dans le chapitre trois ainsi que les algorithmes nous permettent de générer un échantillon de taille  $n$  à partir de ces lois.

#### 3.6.1 Loi Exponentielle

La densité de probabilité d'une loi exponentielle de paramètre  $\lambda > 0$  est donnée par :

$$f(x) = \lambda e^{-\lambda x}, \quad \forall x \geq 0,$$

sa fonction de répartition est définie par :

$$F(x) = 1 - e^{-\lambda x}, \quad \forall x \geq 0.$$

Pour cette distribution on a :

$$\mathbb{E}(X) = \frac{1}{\lambda} \quad \text{et} \quad \sigma^2(X) = \frac{1}{\lambda^2}.$$

**Propriété "sans mémoire" :** Une propriété importante de la loi exponentielle est son absence de mémoire c'est-à-dire,

$$P(X < t + t_0 / X > t_0) = P(X < t), \forall t_0, t > 0.$$

Pour générer une observation d'une loi exponentielle de paramètre  $\lambda$  on exploite l'algorithme suivant :

**Algorithme :**

**Début**

Générer une *v.a*  $U$  uniforme sur  $[0,1]$  ;

$X \leftarrow -\frac{1}{\lambda} \ln(U)$  ;

Retourner  $X$  ;

**Fin ;**

### 3.6.2 Loi d'Erlang

Une variable aléatoire  $X$  suit une loi de Erlang d'ordre  $k$  ( $k = 1, 2, \dots$ ) de paramètre  $\lambda$ , si elle est la somme de  $k$  variables aléatoires indépendantes  $X_1, X_2, \dots, X_k$  et identiquement distribuées suivant une loi exponentielle de paramètre  $\lambda$ . La densité de probabilité d'une loi Erlang d'ordre  $k$  est donnée par :

$$f(t) = \lambda \frac{(\lambda t)^{(k-1)}}{(k-1)!} e^{-\lambda t}, t \geq 0$$

La fonction de répartition est ainsi donnée comme suit :

$$\mathbb{F}(t) = 1 - \int_t^{\infty} f(x)dx$$

Les paramètres  $\lambda$  et  $k$  sont appelés respectivement paramètre d'échelle et de forme.

Pour cette distribution, on a :

$$\mathbb{E}(X) = \frac{k}{\lambda} \quad \text{et} \quad \sigma^2(X) = \frac{k}{\lambda^2}.$$

La loi d'Erlang est un cas particulier de la loi de *Beta*( $a, b$ ) lorsque  $a \in \mathbb{N}$ .

Pour générer une variable aléatoire selon une loi d'Erlang d'ordre  $k$  de paramètre  $\lambda$ , on peut utiliser les algorithmes de génération des nombre aléatoire d'une loi Gamma. Mais sa définition nous permet de construire un simple algorithme de simulation étant donnée une somme  $k$  variable aléatoire d'une loi exponentielles de paramètre  $\lambda$ .

### ***L'algorithme***

#### **Début**

Générer  $k$  v.a uniforme sur  $[0, 1]$  ( $U_1, U_2, \dots, U_n$ );

$$X \leftarrow \sum_{i=1}^n \left(-\frac{1}{\lambda}\right) \log(U_i) = -\frac{1}{\lambda} \log\left(\prod_{i=1}^n U_i\right);$$

Retourner X;

**fin ;**

### 3.6.3 Loi de Weibull

On dit que la variable aléatoire  $X$  suit la loi de Weibull, de paramètres  $(\alpha, \beta, \nu)$ , et on note  $X \rightsquigarrow wbl(\alpha, \beta, \nu)$ , si sa densité s'écrit sous la forme :

$$f(X) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{x-\nu}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x-\nu}{\alpha}\right)^\beta} & \text{si } x \geq \nu \\ 0 & \text{si } x < \nu \end{cases}. \quad (3.2)$$

où :  $\alpha > 0$  est un paramètre d'échelle,  $\beta > 0$  est un paramètre de forme et  $\nu \in \mathbb{R}$  est un paramètre de position. Son espérance et sa variance sont respectivement :

$$E(X) = \alpha \Gamma\left(1 + \frac{1}{\beta}\right) + \nu \quad \text{et} \quad Var(X) = \alpha^2 \left[ \Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right) \right].$$

avec  $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}$ ,  $x \in \mathbb{R}$  (la fonction gamma).

## 3.7 La fonction *RANDOM* sous Matlab

Tous les logiciels de calcul sont munis de fonctions prédéfinies pour la génération de nombre pseudo-aléatoires dont la syntaxe et les lois qui peuvent être simulées diffèrent d'un logiciel à un autre. Sous Matlab on trouve qu'il existe plusieurs fonctions de génération de nombres aléatoires, entre autres on cite : *rand*, *randn* et *random*. La fonction *rand* renvoie des nombres à virgule flottante entre 0 et 1 tiré d'une distribution uniforme, la fonction *randn* est spécifique pour la génération des nombres aléatoires tirés d'une distribution normale centrée et réduite, tandis que la fonction *random* est plus générale où elle peut être utilisée pour générer des variables aléatoires issue d'une loi usuelle fixée préalablement par l'utilisateur. Pour cette dernière, l'utilisateur peut utiliser la commande :

**random(nom,A)** qui renvoie un nombre aléatoire de la famille de distribution à un paramètre spécifiée par "nom" et le paramètre de la distribution *A*.

**Exemple** : `random('exp',2)` rend une observation d'une variable aléatoire de loi exponentielle de paramètre 2.

**random(nom,A,B)** qui renvoie un nombre aléatoire de la famille de distribution ayant deux paramètres, spécifiée par "nom" et les paramètres de la distribution *A* et *B*. **Exemple** : `random('norm',20,2)` rend une observation d'une variable aléatoire de loi normal de moyenne 20 et d'écart-type 2.

**random(nom,A,B,C)** qui renvoie un nombre aléatoire de la famille de distribution à trois paramètres spécifiée par nom et les paramètres de cette distribution *A*, *B* et *C*. **Exemple** : `random('Weibull',2,3,4)` rend une observation d'une variable aléatoire de loi de Weibull de paramètre d'échelle 2, de paramètre de forme 3 et de paramètre de position 4.

**random(...,n<sub>1</sub>,n<sub>2</sub>,...,n<sub>p</sub>)** ou **random(...,[n<sub>1</sub>,n<sub>2</sub>,...,n<sub>p</sub>])** renvoie une matrice de dimension  $n_1 \times n_2 \times \dots \times n_p$  de nombres aléatoires.

**Exemples :**

1. `random('exp',2,5,3)` : rend une matrice de dimension  $5 \times 3$  d'observations issues d'une *v.a* de loi exponentielle de paramètre 2.
2. `random('exp',2,5)` : rend une matrice de dimension  $5 \times 5$  d'observations issues d'une *v.a* de loi exponentielle de paramètre 2.
3. `random('exp',2,5,1)` : rend un vecteur colonne de dimension 5 d'observations issues d'une *v.a* de loi exponentielle de paramètre 2.

Pour plus de détails sur la fonction prédéfinie *RANDOM* et les noms des différentes lois qui peuvent être simulées sous Matlab, le lecteur peut se référer au **HELP** du logiciel Matlab.

## Résumé

Dans ce mémoire nous avons réalisé via l'approche simulation à événements discrets une analyse détaillée d'un système d'attente  $GI/GI/1$  avec Bernoulli feedback de clients dont la probabilité de Feedback dépend du nombre de clients dans le système. Les résultats de simulation obtenus (numérique et graphiques) mettent en évidence principalement l'effet de la distribution des durées des inter-arrivées, la distribution des durées de service, la probabilité de Feedback et l'intensité du trafic sur les caractéristiques stationnaire du système en question.

**Mots clés :** Files d'attente, processus (non)Markovien, feedback, mesures de performance, simulation.

## Abstract

In this thesis, we have carried out via discrete events simulation approach a detailed analysis of a  $GI/GI/1$  queuing system with Bernoulli customer feedback whose feedback probability depends on the number of customers in the system. The simulation obtained results (numerical and graphical) mainly highlight the effect of the distribution of inter-arrivals times, the distribution of service times, the probability of Feedback and the traffic intensity on the stationary characteristics of the system in question.

**Keywords :** Queues, (non)Markovian process, feedback, performance measures, simulation.