

Université de Montréal

**Modélisation et simulation d'un centre d'appels
téléphoniques dans un environnement mixte**

par

Alexandre Deslauriers

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

Mars 2003

©Alexandre Deslauriers, 2002

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

**Modélisation et simulation d'un centre d'appels
téléphoniques dans un environnement mixte**

présenté par:

Alexandre Deslauriers

a été évalué par un jury composé des personnes suivantes:

Michel Gendreau

(président-rapporteur)

Pierre L'Écuyer

(directeur de recherche)

Felisa J. Vázquez-Abad

(membre du jury)

Sommaire

L'objectif de ce mémoire est de construire, analyser et expérimenter plusieurs modèles représentant un centre d'appels téléphoniques opérant en mode mixte, c'est-à-dire avec à la fois des appels entrants et sortants. Ces appels sortants permettent généralement aux agents d'être occupés lorsque le débit des appels entrants est faible. Les propriétés recherchées pour les modèles sont le réalisme et la rapidité d'évaluation de leurs mesures de performance. L'ensemble de ces modèles se basent sur une analyse de données d'un centre d'appels de Bell Canada. Notre but est de trouver un modèle réaliste pour ce centre d'appels nous permettant d'obtenir de bonnes mesures de façon efficace.

Tout d'abord, notre analyse de données tente d'ajuster des lois ou des processus afin d'obtenir une bonne représentation du processus d'arrivée d'appels, des durées de service des appels entrants et sortants, etc. Nous allons appliquer des méthodes d'ajustement connues et en proposer une nouvelle pour un processus d'arrivée particulier. L'ajustement de ce processus se fait à l'aide de la maximisation de sa fonction de vraisemblance. Nous allons vérifier le réalisme de chaque ajustement à l'aide de différents tests statistiques.

Ensuite, nous considérons la simulation pour estimer les mesures de performance d'un modèle complexe et pour étudier la sensibilité du centre d'appels sous différents scénarios (modification de la gestion).

Nous nous intéressons aussi aux modèles de chaînes de Markov en temps continu (CMTC). Nous faisons un résumé des modèles existants dans la littérature et nous proposons des extensions mieux adaptées au centre d'appels étudié. Celles-ci seront utilisées dans la planification du nombre nécessaire d'agents afin de respecter une certaine qualité de service.

Mots-clés : Analyse de données, étude de sensibilité, modélisation, simulation, modèles de CMTC, centre d'appels, environnement mixte.

Abstract

The aim of this thesis is to construct, analyze and experiment several models of a call center operating in a blend environment, i.e., with a mixture of inbound and outbound traffic. The outbound traffic is generally used to increase the agent occupancy. The properties researched in our models are realism and quickness to evaluate their performance measures. We propose models based on a data analysis of a Bell call center. Our goal is to find a realistic model for this call center that can be used to calculate meaningful performance measures.

First, a data analysis tries to fit some probabilistic distributions and processes to various aspects of a call center, such that the arrival process and the service duration of outbound and inbound calls. We apply some known adjustment methods and we propose a new scheme for a particular arrival process. This method finds the parameters that maximize a likelihood function. We check the quality of every adjustment with some statistical tests.

Next, we use a simulation model to obtain performance measures of a complex model and to study the sensitivity of the call center performance with different management policies.

Lastly, we consider some continuous-time Markov chain (CTMC) models. We study the possibility of using models that are available in the literature. Since they are not really applicable to our call center, we propose extensions of these models that are better adapted. We want to use them to find out how a call center should be staffed in order to obtain the desired service quality.

Key words : Data analysis, sensibility analysis, modeling, simulation, CTMC models, call center, blend environment.

Table des matières

1	Introduction	1
1.1	Étude des centres d'appels en mode mixte	1
1.2	Description du modèle général	3
1.3	Fonction objective et contraintes	5
1.4	Projet avec Bell Canada	6
1.5	Bref aperçu des modélisations utilisées dans ce mémoire	8
1.6	Analyse des données recueillies au centre d'appels	11
1.7	La librairie de simulation	12
1.8	Plan du mémoire	12
2	Analyse des données	14
2.1	Les appels entrants	15
2.1.1	Le processus d'arrivée des appels	15
2.1.2	La durée de service des appels entrants	35
2.1.3	Les abandons	41
2.1.4	La qualité de service	44
2.2	Les appels sortants	46
2.2.1	La durée de service des appels sortants	46
2.2.2	Le processus de composition d'appels sortants	52
2.3	Les préposés	55
2.3.1	Le nombre de préposés en fonction	56
2.3.2	Le taux d'occupation des préposés	57

3	Modèle “réaliste” et expérimentations à l’aide de la simulation	64
3.1	Modèle “réaliste”	65
3.1.1	Construction d’un modèle représentant le centre d’appels	68
3.1.2	Simulation du modèle	76
3.1.3	Difficultés rencontrées	88
3.2	Expérimentations	89
3.2.1	Expérimentation #1 : le respect d’une qualité de service à court terme vs à long terme	89
3.2.2	Expérimentation #2 : variation du nombre d’agents disponibles . . .	93
3.2.3	Expérimentation #3 : modifications d’hypothèses dans le modèle . .	97
4	Théorie, extensions et expérimentations sur des modèles de CMTC	100
4.1	File d’attente avec une seule source de clients	101
4.2	Extensions connues dans la littérature	104
4.3	Nouvelles extensions	105
4.3.1	Modèle #1	106
4.3.2	Modèle #2	109
4.3.3	Modèle #3	112
4.4	Expérimentations	116
4.4.1	Mesures de performance pour ces modèles	116
4.4.2	Planification des agents	119
4.5	Extensions possibles	130
5	Conclusion	132
	Bibliographie	134
	Guide d’utilisation d’une librairie de simulation pour centres d’appels	139
	Introduction	140
	Module Distribution	142
	Module Incall	146
	Module Servers	149

Module Waitqueue	152
Module Dialer	154
Module Outcall	156
Module Statistic	158
Module Constant	161
Module Call	163

Liste des figures

1.1	Diagramme du modèle général d'un centre d'appels en mode mixte	3
2.1	Distribution empirique du nombre moyen d'arrivées par demi-heure vs distribution d'une loi de Poisson.	18
2.2	Fonction du taux d'arrivée par demi-heure	20
2.3	Fonction de répartition empirique pour les durées de service d'appels sortants.	47
2.4	Fonction de densité empirique pour les durées de service d'appels sortants.	47
2.5	Fonction de densité empirique pour les durées de service d'appels sortants vs celles de deux lois lognormales.	50
3.1	Processus pour la construction d'un modèle de simulation pour un centre d'appels.	66
3.2	Différence relative entre la valeur empirique et simulée du volume moyen d'appels entrants servis pour chaque demi-heure.	80
3.3	Différence relative entre la valeur empirique et simulée du volume moyen d'appels sortants servis pour chaque demi-heure.	80
3.4	Différence entre le pourcentage empirique et simulé de la qualité de service observée pour chaque demi-heure.	81
3.5	Différence entre le pourcentage empirique et simulé du taux d'occupation des agents pour chaque demi-heure.	81

3.6	Différence relative entre la valeur empirique et simulée du nombre moyen d'abandons pour chaque demi-heure.	82
3.7	Pourcentage simulé d'appels entrants ayant attendu moins de 20 secondes pour chaque demi-heure avec 96% et 100% des agents disponibles.	95
3.8	Nombre moyen d'abandons simulés pour chaque demi-heure avec 96% et 100% des agents disponibles.	95
3.9	Pourcentage simulé d'appels entrants ayant attendu moins de 20 secondes pour chaque demi-heure avec 92,4% et 100% des agents disponibles.	96
3.10	Nombre moyen d'abandons simulés pour chaque demi-heure avec 92,4% et 100% des agents disponibles.	96
4.1	Schéma général d'un processus de naissance et de mort.	102
4.2	La qualité de service (pourcentage d'appels entrants ayant attendu moins de 20 secondes) simulée pour chaque demi-heure du lundi avec la planification actuelle et la planification obtenue à l'aide des modèles de CMTC.	128
4.3	La qualité de service (pourcentage d'appels entrants ayant attendu moins de 20 secondes) simulée pour chaque demi-heure du mercredi avec la planification actuelle et la planification obtenue à l'aide des modèles de CMTC.	128
4.4	Taux d'occupation des agents simulé pour chaque demi-heure du lundi avec la planification actuelle et la planification obtenue à l'aide des modèles de CMTC.	129
4.5	Taux d'occupation des agents simulé pour chaque demi-heure du mercredi avec la planification actuelle et la planification obtenue à l'aide des modèles de CMTC.	129

Liste des tableaux

2.1	Moyenne et variance du nombre d'arrivées par demi-heure pour les jours réguliers de la semaine.	17
2.2	Paramètres et valeurs d'un test statistique pour un processus d'arrivée de Poisson non-homogène spécifique aux lundi, mardi et mercredi.	21
2.3	Paramètres et valeurs d'un test statistique pour un processus d'arrivée de Poisson non-homogène spécifique aux jeudi, vendredi et samedi.	22
2.4	Paramètres estimés et valeurs d'un test statistique avec un processus d'arrivée mixte Poisson-Gamma pour tous les jours et un autre pour les jours de la semaine.	26
2.5	Paramètres estimés et valeurs d'un test statistique avec un processus d'arrivée mixte Poisson-Gamma spécifique aux lundi, mardi et mercredi.	28
2.6	Paramètres estimés et valeurs d'un test statistique avec un processus d'arrivée mixte Poisson-Gamma spécifique aux jeudi, vendredi et samedi.	29
2.7	Estimation des paramètres pour un processus d'arrivée de Poisson corrélé par un facteur de loi gamma.	33
2.8	Moyenne par demi-heure de la durée de service d'un appel entrant.	36
2.9	Valeurs d'un test statistique pour la représentation de la durée de service d'un appel entrant comme une variable aléatoire de loi exponentielle variant par demi-heure.	38

2.10 Paramètres estimés et valeurs d'un test statistique pour la représentation de la durée de service d'un appel entrant comme une variable aléatoire de loi gamma variant par demi-heure.	42
2.11 Intervalle de confiance du pourcentage des appels entrants qui abandonnent avant de se faire servir.	43
2.12 Pourcentage des appels entrants (par demi-heure) ayant attendu moins de 20 secondes avant de se faire servir ou abandonner pour les jours réguliers de la semaine.	45
2.13 Intervalle de confiance sur le nombre d'appels composés, le nombre de clients rejoints, le nombre d'appels nécessitant un temps de service, le nombre de "mismatches" et la proportion d'appels rejoints durant chaque demi-heure d'un jour de semaine.	53
2.14 Intervalle de confiance sur le nombre d'appels composés, le nombre de clients rejoints, le nombre d'appels nécessitant un temps de service, le nombre de "mismatches" et la proportion d'appels rejoints durant chaque demi-heure d'un samedi.	54
2.15 Nombre moyen d'agents disponibles en mode entrant et mixte pour chaque demi-heure des jours de la semaine.	57
2.16 Nombre moyen d'agents disponibles en mode entrant et mixte pour chaque demi-heure du lundi, mardi et mercredi.	59
2.17 Nombre moyen d'agents disponibles en mode entrant et mixte pour chaque demi-heure du jeudi, vendredi et samedi.	60
2.18 Taux d'occupation des agents en mode entrant et mixte pour chaque demi-heure des jours de la semaine.	61
2.19 Taux d'occupation des agents en mode entrant et mixte pour chaque demi-heure du lundi, mardi et mercredi.	62
2.20 Taux d'occupation des agents en mode entrant et mixte pour chaque demi-heure du jeudi, vendredi et samedi.	63

3.1	Paramètres de la loi gamma représentant les durées de service d'appels entrants.	72
3.2	Tableau comparatif entre les mesures de performance obtenues par la simulation de notre modèle mathématique et celles observées dans la réalité. . .	78
3.3	Approximation des p -valeurs de cinq mesures de performance par demi-heure du modèle.	84
3.4	Tableau comparatif entre les mesures de performance obtenues par la simulation des modèles mathématiques du lundi et mardi, et celles observées dans la réalité.	86
3.5	Tableau comparatif entre les mesures de performance obtenues par la simulation des modèles mathématiques du mercredi et jeudi, et celles observées dans la réalité.	87
3.6	Tableau comparatif entre les mesures de performance obtenues par la simulation d'un modèle mathématique pour le vendredi, et celles observées dans la réalité.	87
3.7	Tableau des mesures de performance obtenues par la simulation du modèle R avec un nouveau processus de composition d'appels.	91
3.8	Tableau des mesures de performance obtenues par la simulation du modèle R avec un nouveau processus de composition d'appels.	92
3.9	Tableau des mesures de performance obtenues par la simulation du modèle R avec 96% des agents disponibles.	93
3.10	Tableau des mesures de performance obtenues par la simulation du modèle R avec 92,4% des agents disponibles.	97
3.11	Tableau des mesures de performance obtenues par la simulation du modèle R avec des durées de service exponentielles et un processus d'arrivée stationnaire par demi-heure.	98
3.12	Tableau des mesures de performance obtenues par la simulation du modèle R avec des durées de service exponentielles et un processus d'arrivée Poisson-Gamma.	99

3.13	Tableau des mesures de performance obtenues par la simulation du modèle R avec des durées de service exponentielles.	99
4.1	Paramètres des taux de service utilisés dans les trois modèles de CMTC. . .	121
4.2	Tableau comparatif entre le nombre d'agents utilisé dans la simulation du modèle R et les nombres nécessaires d'agents pour respecter la règle 80-20 sur horizon infini du modèle #1 avec un processus d'arrivée stationnaire et avec un processus d'arrivée Poisson-Gamma.	122
4.3	Tableau comparatif entre le nombre d'agents disponibles dans la simulation du modèle R et le nombre nécessaire d'agents pour respecter la règle 80-20 sur horizon infini sur le modèle #3 avec un processus d'arrivée Poisson-Gamma.	125
4.4	Tableau des mesures de performance obtenues suite à la simulation du modèle R du lundi avec la planification d'agents obtenue à l'aide des modèles de CMTC.	126
4.5	Tableau des mesures de performance obtenues suite à la simulation du modèle R du mercredi avec la planification d'agents obtenue à l'aide des modèles de CMTC.	127

Liste des sigles et abréviations

λ	Taux d'arrivée des appels entrants.
$\lambda(t)$	Fonction du taux d'arrivée des appels entrants variant dans le temps t .
W	Facteur aléatoire associé à une journée qui multiplie la fonction de taux d'arrivée.
μ	Taux de service des appels entrants et sortants.
μ_e	Taux de service des appels entrants.
μ_s	Taux de service des appels sortants.
n	Nombre d'agents disponibles.
\hat{n}	Seuil du nombre d'agents occupés permettant la composition d'appels sortants.
ρ	Intensité du trafic i.e. $\lambda/(n\mu)$.
ν	Taux de connexion d'un client suite à la composition d'un appel sortant.
κ	Probabilité de rejoindre un client à l'aide d'un appel sortant.
(a, b)	Valeurs pour la règle (a, b) de qualité de service. On définit cette règle comme suit : $a\%$ des clients entrants doivent avoir un temps d'attente inférieur à b secondes.
Q	Pourcentage de clients entrants ayant attendu moins de 20 secondes dans la file.
$S_{e,c}$	Durée de service du client entrant c .
$S_{s,c}$	Durée de service du client sortant c .
$A_{e,c}$	Temps maximal d'attente accepté par le client entrant c .
$A_{s,c}$	Temps maximal d'attente accepté par le client sortant c .

H_0	Hypothèse nulle.
TTA	Durée totale de service (“Total Time to Answer”)
CMTC	Chaîne de Markov en temps continu.
EVM	Estimateur de vraisemblance maximale.
EMM	Estimateur par la méthode des moments.
I	Dénote le groupe d’agents pouvant servir uniquement des appels entrants (inbound).
B	Dénote le groupe d’agents pouvant servir des appels entrants et sortants (blend).

Remerciements

Je remercie d'abord Pierre L'Écuyer, mon directeur de travaux, pour l'aide qu'il m'a apportée tout au long de ce travail de maîtrise ainsi que pour m'avoir proposé ce projet intéressant.

Je remercie également les gens de Bell Canada, pour avoir eu la patience de répondre à mes questions et pour avoir fait leur possible afin d'améliorer ma compréhension sur plusieurs détails du centre d'appels. Je tiens à remercier principalement Raphaël Bean, Michel Houle, Frédéric Forget et Ginette Côté.

Ensuite, je tiens à remercier Richard Simard, Nabil Channouf, Athanassios Avramidis et Jutta Pichitlamken pour vos conseils et nos quelques échanges d'idées. Et je leur souhaite bonne chance dans la suite du projet.

Je suis reconnaissant envers le CRSNG pour son support financier tout au long de ma maîtrise.

Je veux aussi souligner ma reconnaissance envers Francis Picard qui a été une source de dépassement tout au long de mes études universitaires.

Et finalement, je veux remercier mes parents, Henri et Hélène, qui ont pris soin que j'obtienne une bonne éducation sans laquelle la réalisation de ce mémoire n'aurait été possible.

Chapitre 1

Introduction

Les centres d'appels prennent, depuis un certain temps, une importance capitale dans toute grande entreprise moderne. Ils sont des outils efficaces pour offrir des services et pour communiquer avec les clients. On estime qu'aux États-Unis, de 4 à 7 millions de personnes, selon les sources, travaillent dans environ 70000 centres d'appels qui génèrent des revenus de plusieurs dizaines de milliards de dollars par année [15, 2, 49, 32]. Vu l'importance des sommes d'argent impliquées, l'étude de ces centres d'appels afin d'optimiser leur gestion est amplement justifiée.

1.1 Étude des centres d'appels en mode mixte

La plupart des centres d'appels se doivent de respecter une certaine qualité de service. Le temps d'attente et les abandons de clients sont souvent les critères qui composent cette qualité. Les critères peuvent être définis par un organisme extérieur ou par l'entreprise elle-même. Ainsi, le centre d'appels doit adapter sa gestion afin de respecter la qualité de service requise. Le nombre d'agents travaillant dans le centre font partie de ce qu'il faut gérer et constituent l'une des principales sources de coûts dans un centre d'appels. En fait, ils en représentent à peu près les trois quarts [35]. Une utilisation optimale de ceux-ci, tout en respectant la qualité de service, permet de générer un maximum de revenus, en supposant bien sûr que l'augmentation de l'occupation d'un agent implique des revenus supplémentaires. Une manière toute simple d'augmenter celle-ci est d'affecter les agents à d'autres tâches lorsque le trafic d'arrivée d'appels est bas. Ces tâches peuvent être de

répondre à des courriels, à des “fax”, à des clients rejoints par le centre, etc.

Dans ce mémoire, il est question des centres d’appels qui tentent d’augmenter cette occupation en opérant en mode mixte, i.e., avec à la fois des appels entrants initiés par les clients et des appels sortants initiés par le centre. Le centre n’a pas le contrôle sur les appels entrants, mais il gère les appels sortants à l’aide d’un système de composition automatique. Ces appels sont généralement initiés lorsque le trafic des appels entrants est trop faible pour maintenir un taux d’occupation des agents acceptable. Le but de ces appels peut être, par exemple, de rappeler certains clients qui ont laissé un message, faire du marketing, etc.

Les politiques de gestion se comparent entre elles en étudiant la “performance” qu’elles procurent au centre d’appels. Les mesures de performance pourraient être le trafic d’appels sortants servis, le taux d’occupation des agents ou le salaire versé aux préposés. Il est important de noter que les politiques de gestion doivent être acceptables, i.e., elles doivent respecter une certaine qualité de service fixée par le centre. L’objectif visé dans ce centre d’appels est d’utiliser une politique acceptable qui maximise sa performance. Le besoin d’évaluer les mesures caractérisant la performance d’une politique nous force à construire des modèles mathématiques représentant le centre d’appels. Tout dépendant du niveau de complexité désiré, nous pouvons faire l’étude des centres d’appels à l’aide de modèles de chaînes de Markov en temps continu (CMTC) [10, 11, 13, 12], de la simulation [14, 37, 49], de la programmation dynamique stochastique [1, 9, 10, 23] et de l’exploitation de données (“data-mining”). Dans ce mémoire, nous nous concentrerons exclusivement sur les deux premières méthodes.

Les modèles de CMTC auront l’avantage de fournir rapidement des approximations de la performance d’une politique de gestion. Ces mesures de performance seront trouvées à l’aide de formules analytiques ou de calculs numériques. Cependant, les CMTC ont le désavantage d’utiliser des hypothèses fortes. D’un autre côté, la simulation permet de traiter des modèles mathématiques plus complexes et d’éliminer plusieurs hypothèses fortes faites avec les CMTC. L’estimation des mesures de performance sera possible en simulant, sur plusieurs jours, le modèle construit. La simulation permettra de trouver assez rapidement des mesures précises sous forme d’intervalles de confiance.

Le but de ce mémoire est de construire des modèles mathématiques représentant le mieux possible un centre d’appels en mode mixte. Ces modèles seront utiles pour estimer

des mesures de performance dont la qualité de service et le trafic d'appels sortants servis. Les mesures obtenues nous permettront d'évaluer le réalisme de chacun des modèles. Nous allons considérer qu'un modèle est réaliste s'il présente des mesures d'intérêt semblables à celles que l'on retrouve dans la réalité. Plus la différence sera petite entre celles-ci, plus le modèle sera jugé réaliste. La vitesse de calcul, la précision et le réalisme des mesures de performance seront les critères retenus pour comparer nos modèles. Par la suite, ces modèles seront utilisés pour étudier le comportement du centre sous différents scénarios ou politiques de gestion.

1.2 Description du modèle général

Le modèle général que nous allons présenter tente de représenter les comportements observés dans un centre d'appels de type mixte. La figure 1.1 donne une représentation graphique du modèle.

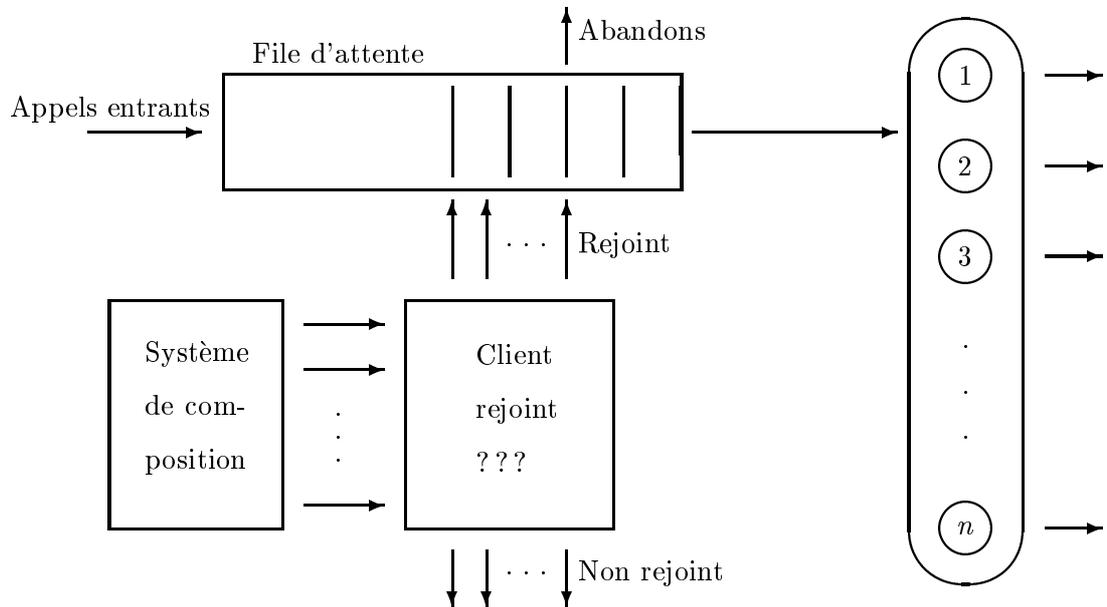


Figure 1.1 – Diagramme du modèle général d'un centre d'appels en mode mixte

Le centre d'appels opère dans un environnement incertain. Les appels entrants arrivent de manière aléatoire. Le trafic observé pour les arrivées d'appels entrants peut dépendre de la journée (lundi, mardi, ...), de la température, de l'heure, etc. Chaque client doit entrer

dans la file d'attente à capacité infinie de type FIFO ("First In First Out"). Si la file est vide et qu'il y a au moins un agent inoccupé alors le client débute immédiatement son service avec un des agents libres. Si ce n'est pas le cas, alors le client doit attendre qu'un serveur se libère. Son service débutera dès qu'il sera le client ayant attendu le plus longtemps et qu'il y aura au moins un agent inoccupé. Il peut y avoir des abandons, c'est-à-dire qu'un client se décourage et décide de quitter sans recevoir son service. Le temps maximal d'attente accepté d'un client est défini par la durée entre l'instant où le client débute son attente dans la file et l'instant où il quitte. Cette durée est aléatoire.

Les appels sortants sont initiés par un système de composition automatique. Le nombre d'appels sortants à composer en parallèle se base sur différents facteurs. Par exemple, il peut dépendre du nombre d'agents libres ou de la distribution du temps d'attente par client depuis un certain temps. Parmi les appels composés, il y en a un certain nombre qui seront sans réponse (client absent, ligne occupée,...). Les clients rejoints sont insérés dans la file et servis immédiatement si possible. Si le service d'un client sortant ne peut débuter immédiatement alors on le comptabilise comme un "mismatch". Le temps maximal d'attente accepté par un client sortant a la même définition que pour le client entrant. Cette durée est aléatoire et peut avoir une loi différente de celle d'un client entrant.

Le centre d'appels est composé d'un ensemble de n préposés ou agents. Ces préposés ont comme tâche de servir les clients. Dans notre modèle, on suppose qu'ils sont identiques, mais ils n'ont pas tous les mêmes tâches. Certains serviront uniquement des appels entrants tandis que d'autres serviront autant les appels sortants qu'entrants. Cette hypothèse nous permet de les modéliser en deux groupes au lieu de les voir individuellement. La durée requise à un agent pour servir un client est aléatoire. De plus, la loi représentant la durée de service d'un appel sortant peut être différente de celle d'un appel entrant. Une fois le service d'un client terminé, le client quitte le centre, et le préposé l'ayant servi devient libre. Il est alors prêt à servir un autre client.

Ces événements peuvent survenir à un instant t réel (temps continu). La composition des appels sortants se fait généralement lorsque le service d'un client se termine.

Dans ce mémoire, nous allons proposer différentes modélisations qui se baseront sur ce modèle général. Elles seront caractérisées par différents degrés de réalisme et d'utilité (précision des estimations, vitesse de calcul, ...).

1.3 Fonction objective et contraintes

Une bonne gestion d'un centre d'appels général a pour objectif d'assurer une bonne qualité de service au moindre coût. Pour les centres en mode mixte, on peut minimiser les coûts en réduisant le nombre nécessaire d'agents (le coût relié aux salaires diminue) ou en augmentant le volume d'appels sortants (le profit relié aux appels sortants augmente). Chacune de ces optimisations est contrainte par une qualité de service. Celle-ci peut se mesurer par des aspects qualitatifs (le client est-il satisfait de la réponse obtenue, l'agent a-t-il proposé les meilleurs services au client, etc.) ou par des aspects quantitatifs (quel fut le temps d'attente du client, a-t-on répondu au client avant qu'il ne raccroche, etc.). Dans ce mémoire, nous allons nous intéresser uniquement aux mesures quantitatives.

Ces mesures de qualité s'expriment souvent en termes de temps d'attente et de taux d'abandon pour les clients d'appels entrants. Ce taux d'abandon est défini par la fraction des appels entrants qui quittent la file d'attente sans avoir obtenu un service. Les critères de temps d'attente peuvent être la moyenne d'attente par client ou être la règle (a, b) . On définit la règle (a, b) comme suit : $a\%$ des clients entrants doivent avoir un temps d'attente inférieur à b secondes. Les valeurs fréquemment utilisées sont $a = 80$ et $b = 20$ ou encore, $a = 70$ et $b = 30$. Une contrainte peut être interprétée ou exigée de différentes manières. On peut, par exemple, l'imposer en moyenne sur une année, sur un mois, sur une journée, sur une heure, etc. Il est assez facile de voir qu'en imposant une contrainte en moyenne sur une longue période, celle-ci devient beaucoup moins restrictive que si on l'impose en moyenne sur une courte période. Une moindre restriction peut permettre d'améliorer la performance du centre d'appels. Un autre type de contrainte est d'imposer une borne supérieure sur le taux d'abandon dans la file d'attente. Encore une fois, nous pouvons tenter de satisfaire cette contrainte en moyenne sur différentes périodes de temps. Le taux d'abandon est bien évidemment lié à la distribution du temps d'attente. Une augmentation du temps d'attente signifie qu'un plus grand nombre de clients vont dépasser leur seuil d'attente acceptée et ainsi, le taux d'abandon va lui aussi augmenter. La plupart du temps, une bonne qualité de service au niveau du temps d'attente implique par le fait même une bonne qualité de service au niveau des abandons.

Dans les centres d'appels en mode mixte, nous rencontrons aussi des contraintes sur les appels sortants. Dans ce type de centre, les appels sortants sont initiés par un système de

composition. Il peut se produire qu'il soit impossible de servir immédiatement les clients rejoints à cause d'un manque d'agents libres. Cette pénurie d'agents peut être attribuée à une augmentation subite du trafic d'appels entrants. Ce type de situation que l'on nomme "mismatch" nuit à la qualité de service. Tout dépendant des centres, il se peut qu'on impose une borne supérieure sur le pourcentage de "mismatches", i.e., le nombre de "mismatches" par appel sortant rejoint. Cette contrainte peut être imposée en moyenne sur différentes périodes de temps.

Nous venons de donner une idée des contraintes que l'on peut exiger dans un centre d'appels en général. Les contraintes spécifiques au centre d'appels étudié dans ce mémoire seront décrites plus loin.

1.4 Projet avec Bell Canada

Ce mémoire a été fait dans le cadre d'un projet au GERAD (Groupe d'Études et de Recherche en Analyse des Décisions), intitulé *Gestion optimale d'un centre d'appels dans un environnement "blend"*. Ce projet vise à optimiser les politiques de gestion d'un centre d'appels de Bell Canada. Ce centre permet aux clients d'avoir accès directement à une foule de renseignements, de produits et services de Bell Canada, Bell Mobilité, Bell Express Vu et du service Internet Sympatico.

Les heures d'ouverture de ce centre sont de 8h à 20h30 du lundi au vendredi et de 8h30 à 17h le samedi. Le nombre de préposés dans le centre varie selon les horaires journaliers définis par une équipe de planificateurs. Un agent travaille durant sept heures et demie. Règle générale, l'agent travaille les premières quatre heures et demie en mode entrant (fait seulement le service des appels entrants) pour ensuite continuer en mode mixte (entrants et sortants) jusqu'à la fin de sa journée de travail. Le nombre d'agents en service au courant d'une journée varie selon les pauses café, les heures de repas, les réunions de travail, les séances de formation, etc. En moyenne, le nombre de préposés en service est d'environ 26,7 pour les jours de semaine et de 13,3 pour la journée du samedi.

Les appels entrants arrivent de façon aléatoire. Ils sont initialement traités par le IVR ("Interactive Voice Response"). L'IVR a pour tâche d'acheminer les appels vers le centre associé au service désiré ou au type de client. Le centre d'appels étudié s'occupe de plusieurs

clients provenant autant du Québec que de l'Ontario. Les appels entrants sont habituellement initiés par des clients ayant des questions sur leurs services et/ou ayant été rejoints suite à une campagne de marketing.

Les appels sortants sont initiés par le système de composition Mosaix qui est développé par la compagnie Avaya [3]. L'ensemble des clients à rejoindre se base sur des listes fournies par le département de marketing. Pour connaître le nombre d'appels sortants devant être composés à chaque instant, le système de composition se base sur la qualité de service des appels entrants et sur le fait qu'il y ait au moins un préposé inoccupé. La qualité de service est mesurée par un autre système nommé MAX. Ce dernier informe Mosaix de la qualité au cours des 10 dernières minutes. Dès qu'elle est inférieure à un certain seuil, on ne fait aucun appel sortant. En plus de Mosaix, la composition des appels est contrôlée par des gestionnaires d'accès. Ces employés doivent limiter les pertes de temps des préposés, obtenir une bonne qualité de service quotidienne et obtenir un bon taux d'occupation des préposés. Les deux dernières tâches sont les plus intéressantes pour nous. Pour corriger la performance dans le centre, le gestionnaire d'accès dispose d'un ensemble d'actions dont l'annulation de réunions pour augmenter les effectifs, la modification des paramètres du système Mosaix et la prise de messages pour des appels en attente. Par exemple, s'il se rend compte que sa qualité de service est mauvaise pour les deux dernières demi-heures, il va alors diminuer l'agressivité du système Mosaix. En bref, la tâche du gestionnaire est d'atteindre les objectifs fixés par le centre d'appels qui sont d'avoir au minimum une qualité de service 80-20, i.e., avoir au moins 80 % des clients d'appels entrants qui attendent moins de 20 secondes.

Le centre d'appels que nous venons de décrire est un cas particulier du modèle général introduit à la section 1.2. Nous allons utiliser des données recueillies à ce centre pour construire des modélisations mathématiques qui le représentent bien. Par la suite, elles seront utilisées afin d'évaluer et de développer des politiques de gestion tant au niveau de la composition des appels sortants qu'au niveau de la planification du nombre nécessaire d'agents.

Comme il en fut question plus haut, ce mémoire fait partie du projet de recherche *Gestion optimale d'un centre d'appels dans un environnement "blend"*. Les travaux serviront de base à la suite du projet dont l'objectif final est de fournir à Bell Canada des outils pour une meilleure gestion de leurs centres d'appels.

1.5 Bref aperçu des modélisations utilisées dans ce mémoire

Dans la section 1.1, nous avons introduit quelques modélisations possibles pour représenter un centre d'appels en mode mixte. Dans ce mémoire, nous avons décidé de nous concentrer sur seulement deux types de modèles, soit les modèles de CMTC et les modèles de simulation.

La simulation est un outil intéressant pour obtenir les mesures de performance d'un modèle complexe. De plus, elle permet d'obtenir une estimation précise de la performance d'un centre d'appels sous différentes politiques de gestion ou scénarios. Tout dépendamment de la complexité du modèle, cette estimation peut être faite dans un temps de calcul relativement rapide. L'utilisation de la simulation est bien répandue dans la communauté scientifique. Les références [4, 5, 14, 37, 44] sont des livres d'introduction à la simulation. Elles sont utiles pour savoir comment construire correctement un modèle de simulation, obtenir l'estimation des mesures de performance et tout autres informations sur la simulation en général.

Nous allons maintenant introduire quelques articles utilisant la simulation pour l'étude des centres d'appels. Klungle [33] montre l'utilité de la simulation dans le processus de gestion d'un centre d'appels fonctionnant en mode entrant seulement. La planification du nombre nécessaire d'agents se fait historiquement par l'entremise d'un modèle analytique comme celui dans la formule Erlang-C [19]. L'auteur présente les forces et les faiblesses d'un modèle analytique et d'un modèle de simulation. Cette analyse est supportée par une application dans un centre de réclamations. Klungle note que les modèles analytiques peuvent donner de grandes erreurs de prévisions. Ceci est dû à quelques hypothèses trop fortes. Parmi celles-ci, il y a la supposition que les arrivées d'appels suivent un processus de Poisson homogène. Ces erreurs peuvent mener à une mauvaise planification du nombre nécessaire d'agents. L'auteur termine son article en énumérant les caractéristiques des applications devant être modélisées par un modèle analytique et celles devant utiliser la simulation.

L'article de Bapat et Pruitte [6] poursuit avec la même idée en expliquant les raisons d'utiliser la simulation pour les centres d'appels d'aujourd'hui. La grande complexité (priorité d'appels, interactions entre les centres, différents types de clients, ...) des centres d'appels est la principale cause évoquée. L'article fait référence au modèle analytique Erlang-C qui sur-planifie le nombre nécessaire d'agents. Ceci vient principalement du fait qu'il n'y a pas

d'abandons dans la file. L'utilisation de la simulation est justifiée par le simple fait que cette mauvaise planification peut engendrer des coûts d'opération inutiles. Les auteurs terminent leur article en citant un exemple d'application où la simulation fut d'une grande utilité pour comparer différentes gestions d'un centre d'appels.

Les trois prochains articles traitent de la construction d'un modèle de simulation pour des centres d'appels complexes. Chokshi [17] traite d'un processus de gestion, utilisant la simulation, qui a été développé chez AT&T. Ensuite, Tanir et Booth [51] présentent les étapes suivies pour construire et ajuster un modèle de simulation pour un centre d'appels de Bell Canada. Le but est de réussir à simuler l'expérience d'un client du moment où il est pris en charge par l'IVR ("Integrated Voice Response") jusqu'à la fin de son service. Cet article explique les difficultés rencontrées lors de la modélisation et donne des solutions potentielles. Les auteurs parlent des problèmes avec la cueillette de données. Ils comparent la précision des mesures de performance en se basant sur des données par 30 minutes vs 5 minutes. Les auteurs montrent que les données par intervalle de 5 minutes permettent de simuler un volume d'arrivées des appels par demi-heure plus près de ce qu'on observe dans la réalité. Finalement, Tanir et Booth démontrent la capacité de la simulation à faire des expérimentations sur différents scénarios possibles : volume d'appels anormaux, nouvelle campagne "marketing", variation dans la planification du nombre d'agents,... Ensuite, il y a l'article de Gulati [26] qui utilise la simulation pour améliorer la gestion d'un centre d'appels. Toutefois, ce centre a la particularité de servir uniquement des appels sortants. Il est intéressant de s'informer sur un tel modèle, car nous allons devoir représenter le processus d'arrivée des appels sortants pour notre centre d'appels.

Dans ce mémoire, nous allons proposer des modèles de CMTC et des modèles plus complexes (simulation). Nous savons bien qu'ils ont chacun leur avantage. Il est clair que nous ne pouvons pas dire qu'un tel modèle est le plus utile dans 100 % des applications. L'article de van Dijk [53] traite de la combinaison des deux techniques, i.e., formule analytique et simulation. L'auteur explique que l'on doit choisir la meilleure méthode selon l'évaluation que l'on veut obtenir. L'étude de cette méthode hybride est intéressante, car nous allons développer les deux types de modèles.

La quantité d'articles sur le sujet est assez importante. On pourrait ajouter l'article de Henderson [27] qui utilise la simulation pour calculer le personnel requis dans un centre d'appels en mode entrant ou encore, celui de Saltzman et Mehrotra [49] qui s'en sert pour

évaluer l'utilité de l'ajout d'un service payant dans un centre d'appels de support technique.

Tous les articles cités se rejoignent sur le point que la simulation permet une grande flexibilité pour représenter un centre d'appels. Les centres d'appels en mode mixte sont une extension des centres d'appels en mode entrant et vu la rigidité des modèles analytiques, il semble que la simulation se prête bien mieux à notre type de centre d'appels.

Le deuxième type de modèles utilise les CMTC. Ceux-ci ont retenu grandement l'attention et sont le sujet d'un imposant nombre de publications. Les CMTC englobent bien évidemment les modèles de files d'attente du type $M/M/n$, $G/G/n$, $M(t)/M(t)/n$, etc. Plusieurs résultats populaires sont connus pour ces modèles, dont toute la famille des formules d'Erlang qui permettent de calculer la distribution du temps d'attente des clients sur horizon infini.

On peut modéliser un centre d'appels fonctionnant seulement en mode entrant comme une file $M/M/n$ de capacité infinie. Dans ce modèle, nous devons faire l'hypothèse que nous avons un processus de Poisson stationnaire pour les arrivées des appels. Nous supposons aussi des durées de service indépendantes suivant une loi exponentielle. Ce modèle comporte plusieurs lacunes pour modéliser un centre d'appels en mode mixte. Par exemple, il ne tient pas compte du fait qu'il y a des appels sortants. De plus, l'hypothèse que les arrivées suivent un processus de Poisson stationnaire est souvent trop forte. Dans ce mémoire, nous allons présenter des extensions du modèle de file $M/M/n$ qui permettent d'améliorer la modélisation de notre centre d'appels.

Dans un premier temps, nous avons construit des CMTC stationnaires, i.e., avec un processus de Poisson homogène pour l'arrivée des appels entrants et une distribution exponentielle stationnaire pour la durée de service des appels entrants ou sortants. La grande différence avec la file $M/M/n$ est l'ajout d'un traitement pour les appels sortants. Nous allons présenter trois de ces modèles qui présenteront différents niveaux de détails sur le centre d'appels. L'avantage d'utiliser un modèle plus complexe est qu'on va obtenir des mesures de performance plus réalistes. Cependant, ceci implique souvent une augmentation du temps de calcul pour obtenir ces mesures. Certains modèles permettront l'utilisation de formules analytiques simples tandis que d'autres vont nécessiter une résolution numérique plus complexe. Nous devons trouver le juste milieu entre le réalisme et la rapidité de calcul. Ces modèles évoluant dans un environnement stationnaire seront utiles dans l'évaluation

du comportement demi-heure par demi-heure, car nous allons supposer que le système est stationnaire durant chaque demi-heure. Nous allons aussi étudier la possibilité d'ajouter plus de variabilité dans le nombre d'arrivées par demi-heure.

Dans le chapitre 4, nous allons présenter les détails sur des modèles de CMTC que l'on retrouve dans la littérature. Ensuite, nous allons présenter nos nouvelles CMTC appliquées au centre d'appels en mode mixte ainsi que des résultats d'expérimentations.

1.6 Analyse des données recueillies au centre d'appels

Les modèles mathématiques proposés comportent différents paramètres à fixer. Cette étape nécessite une cueillette et une analyse des données sur le centre d'appels étudié. De plus, cette analyse sera utile lors de la comparaison entre les mesures de performance d'un modèle et celles observées dans la réalité. Nous pourrions, par exemple, comparer les trafics d'appels sortants servis, les taux d'occupation des préposés, etc.

Les données que nous avons utilisées pour faire cette analyse proviennent d'un centre d'appels de Bell Canada (voir section 1.4). L'analyse des données nous permet de trouver des lois pour les aspects aléatoires du centre d'appels. À l'aide de ces lois, nous allons pouvoir modéliser les instants d'arrivée des appels entrants, les durées de service des appels entrants, les durées de service des appels sortants, la durée maximale d'attente acceptée, etc. Le réalisme de ces lois sera évalué par des tests statistiques. Il est important de noter que l'analyse des données ne sera pas aussi approfondie que l'on aurait voulu. Ceci est dû au fait que les données pertinentes sont souvent agrégées par demi-heure, parfois incomplètes ou tout simplement absentes.

Les détails sur notre analyse des données sont présentés dans le chapitre 2. Nous allons présenter le contenu de la base de données fournie par Bell Canada, les modélisations des aspects stochastiques, les tests statistiques et un résumé sur certaines informations (taux d'occupation des agents, volume d'appels sortants, niveau de la qualité de service, ...) concernant le centre.

1.7 La librairie de simulation

Afin de faciliter la construction des différents modèles de simulation pour un centre d'appels en mode mixte, nous avons développé une librairie logicielle pour la simulation de centre d'appels. La librairie modélise un centre d'appels en modules distincts. Par exemple, nous avons défini un module pour le groupe de préposés, un autre pour l'arrivée des appels entrants, etc. La construction des modèles se fait en définissant le comportement de chaque module et les interactions entre eux. La division en modules permet une bonne flexibilité au niveau de la modélisation.

De plus, il y a un module recueillant des statistiques. Ce module permet de calculer des mesures de performance pertinentes à l'évaluation d'un modèle soumis à une politique de gestion précise. Ces estimations sont obtenues suite à la simulation du modèle sur plusieurs journées. Leurs calculs se font rapidement, car la librairie est programmée en langage C et utilise la librairie de simulation SSC [40], axée sur l'efficacité. La librairie SSC est elle aussi implémentée en C. Nous avons inclus en annexe de ce mémoire un guide sur notre librairie.

1.8 Plan du mémoire

La suite du mémoire est construite comme suit. Le deuxième chapitre présente une analyse des données du centre d'appels étudié. Nous utilisons cette analyse pour ajuster les paramètres des modèles représentant un centre d'appels en mode mixte. Le chapitre 3 présente un modèle que nous avons construit pour la simulation. Nous y présentons les démarches faites pour l'obtenir ainsi que des expérimentations pertinentes. Ces expériences serviront surtout à évaluer la sensibilité du comportement du centre d'appels sous différents scénarios.

Dans le chapitre 4, nous allons présenter la théorie et les publications existantes sur les centres d'appels modélisés par des modèles de CMTC. Comme il n'y en a pas qui s'applique bien à notre centre d'appels, nous allons proposer des nouveaux modèles de CMTC qui le représentent plus fidèlement. Ensuite, nous allons présenter des résultats d'expérimentations utilisant ces modèles.

Finalement, la conclusion permettra de résumer la contribution de ce mémoire et l'utilité

des modèles construits pour la suite du projet.

Chapitre 2

Analyse des données

La modélisation générale d'un système se fait souvent en deux étapes. Premièrement, nous devons comprendre son fonctionnement global. Dans notre centre d'appels, les priorités d'appels et l'algorithme pour la composition des appels sortants sont des exemples de connaissances que l'on doit acquérir. La plupart du temps, il est possible d'obtenir ces informations par l'entremise d'experts. Ensuite, la deuxième étape consiste à modéliser les différentes parties aléatoires. Dans notre cas, il s'agit des arrivées d'appels entrants, des durées de service, du succès de la composition d'appels sortants, etc. Ceci est possible à l'aide d'une analyse des données recueillies dans un centre d'appels réel. Tout dépendant de l'ampleur de l'étude, elle nous permettra d'obtenir un modèle plus ou moins réaliste. Dans la majorité des cas, nous allons tenter de représenter chacun de ces comportements comme un processus stochastique. Ce chapitre sera exclusivement consacré à cette étape de la modélisation. Les résultats obtenus permettront de fixer les paramètres des modèles proposés.

Nous allons analyser les aspects du centre d'appels à l'aide de plusieurs bases de données fournies par Bell Canada. Dans celles-ci, nous retrouvons des informations sur les appels entrants, les appels sortants et les préposés. Ce chapitre consacre une section sur l'analyse de chacun de ces trois aspects.

2.1 Les appels entrants

Dans cette section, nous allons modéliser les aspects touchant les appels entrants. Nous allons nous attarder sur le processus d'arrivée des appels entrants, la durée de service des appels entrants, le processus d'abandon dans la file d'attente et la qualité de service associée à ces appels. Les données fournies par Bell regroupent 253 jours entre le 2 mai 2001 et le 3 mars 2002. Comme nous allons le voir, le raffinement des modélisations proposées sera limité par le fait que nous avons des observations agrégées par demi-heure.

2.1.1 Le processus d'arrivée des appels

Le développement d'un processus d'arrivée des appels réaliste n'est pas une tâche facile. Nous avons plusieurs variantes disponibles pour en faire la modélisation. Par exemple, nous pouvons supposer qu'il dépend ou non de variables extérieures. Ces variables peuvent être l'heure de la journée, la journée (lundi, mardi,...), les conditions climatiques (pluie, tempête de neige, ...), etc. Il est clair que ces variables ont un différent niveau d'influence sur les arrivées d'appels. Nous devons choisir celles qui ont le plus d'impact.

Dans un premier temps, nous allons étudier le cas simple où les arrivées suivent un processus de Poisson avec un taux λ stationnaire ([52] p.270), i.e., nous ne considérons aucune variable extérieure. Nous allons dénoter le taux d'arrivée par demi-heure par le paramètre λ . Sous l'hypothèse d'un processus de Poisson, nous savons que le nombre d'arrivées X pour chaque demi-heure suit une loi de Poisson de taux λ définie par la fonction de masse suivante [52] :

$$\Pr(X = x) = \frac{e^{-\lambda} \lambda^x}{x!} \text{ pour } x = 0, 1, \dots \quad (2.1)$$

Parmi les données disponibles, nous avons le nombre d'arrivées x_k pour chaque demi-heure k . En utilisant le résultat (2.1), il ne reste plus qu'à trouver un estimateur $\hat{\lambda}$ qui ajuste bien la loi de Poisson à tous les x_k arrivées observées. Ce $\hat{\lambda}$ sera considéré comme étant le meilleur estimateur pour le λ du processus de Poisson stationnaire. Nous pouvons calculer l'estimateur $\hat{\lambda}$ à l'aide de la méthode de vraisemblance maximale (EVM) [37]. Celle-ci nous indique que $\hat{\lambda}$ est la moyenne empirique du nombre d'arrivées observées par demi-heure. Avec un total de d demi-heures, nous obtenons :

$$\hat{\lambda} = \frac{\sum_{k=1}^d x_k}{d}. \quad (2.2)$$

Parmi les 253 jours, nous avons 209 jours de semaine (25 demi-heures par jour) et 44 samedis (17 demi-heures par jour). La valeur de $\hat{\lambda}$ trouvée est 44,3. Maintenant, il reste à savoir si un processus de Poisson avec un taux de 44,3 arrivées par demi-heure est réaliste. Sous l'hypothèse décrite par (2.1), on devrait obtenir une variance dans le nombre d'arrivées qui est égale à sa moyenne. Nous connaissons la moyenne empirique et il est possible de calculer un estimateur \hat{s}^2 sans biais de la variance du nombre d'arrivées par demi-heure grâce à cette formule :

$$\hat{s}^2 = \frac{1}{d-1} \sum_{k=0}^d (x_k - \hat{\lambda})^2, \quad (2.3)$$

Nos calculs nous donnent $\hat{s}^2 = 417,8$. Ainsi, nous obtenons une estimation de la variance très loin de la moyenne. Ceci nous indique que les arrivées ne suivent pas un processus de Poisson stationnaire. De plus, la figure 2.1 nous montre bien que la distribution empirique du nombre d'arrivées par demi-heure est différente de la distribution de la loi de Poisson avec un taux $\hat{\lambda} = 44,3$. L'observation de la moyenne d'arrivées par demi-heure nous indique que celle-ci semble être influencée par l'heure de la journée (voir tableau 2.1). Tous ces indicateurs nous montrent qu'un processus stationnaire ne s'applique pas à notre centre d'appels.

Pour se convaincre que cette modélisation n'est pas la bonne, nous pouvons faire le test du chi-deux de Pearson [45]. Ce dernier fait partie de la famille des tests d'ajustement "goodness-of-fit tests" qui sont utilisés pour tester l'hypothèse nulle suivante :

H_0 : Les observations X_i sont des variables i.i.d (indépendantes et identiquement distribuées) provenant de la fonction de distribution \hat{F} .

Dans notre cas, les X_i sont le nombre d'arrivées observées par demi-heure et \hat{F} est la loi de Poisson avec un taux de $\hat{\lambda}$. Les détails concernant cette famille de test se retrouvent dans [37] (p. 356-370). Le test de chi-deux fait partie de cette famille et permet d'obtenir la statistique X^2 qui représente une mesure de distance entre la loi empirique du nombre d'arrivées par demi-heure et la loi de Poisson avec le paramètre $\hat{\lambda}$. Nous allons décrire ce test puisqu'il sera utilisé par la suite. Afin d'utiliser ce test, nous avons divisé la plage des nombres d'arrivées observées en $m = 42$ intervalles. La fréquence observée (O_k) pour un intervalle k est le nombre de demi-heures où le nombre d'arrivées est contenu dans cet intervalle. Ensuite, nous définissons la fréquence espérée par $E_k = d * \Pr(X \in k)$, où d est le nombre de demi-heures et $\Pr(X \in k)$ est la probabilité que le nombre d'arrivées soit

Demi-heure i	$\hat{\lambda}_i$	\hat{s}_i^2	p_i
8h00-8h30	26,1	70,3	9,9e-13
8h30-9h00	38,3	82,3	8,9e-13
9h00-9h30	50,9	142	6,5e-13
9h30-10h00	58	168	6,1e-13
10h00-10h30	61,4	222	7,5e-13
10h30-11h00	62,2	200	7,4e-13
11h00-11h30	62,6	227	7,4e-13
11h30-12h00	60,5	234	8,9e-13
12h00-12h30	56,8	220	7,9e-13
12h30-13h00	56,7	196	9,9e-13
13h00-13h30	59	275	6,1e-13
13h30-14h00	57,3	245	6e-13
14h00-14h30	56,5	200	9,5e-13
14h30-15h00	55,4	215	7,8e-13
15h00-15h30	56,5	189	9,8e-13
15h30-16h00	59	209	6,3e-13
16h00-16h30	59,9	198	6,8e-13
16h30-17h00	54,7	187	7,9e-13
17h00-17h30	43,5	155	9,5e-13
17h30-18h00	40	126	8,7e-13
18h00-18h30	33,3	109	6,3e-13
18h30-19h00	29,7	98,6	7,2e-13
19h00-19h30	25	80,8	8,7e-13
19h30-20h00	22,5	58,6	5,3e-13
20h00-20h30	17,9	44,3	6,5e-13

Tableau 2.1 – Moyenne, dénotée par $\hat{\lambda}_i$, du nombre d'arrivées dans la demi-heure i . La variance du nombre d'arrivées par demi-heure est donnée par \hat{s}_i^2 . Nous donnons aussi la p -valeur du test de chi-deux. Ces valeurs ne tiennent pas compte de la journée du samedi ainsi que de 5 journées dont les heures d'ouverture ont été modifiées.

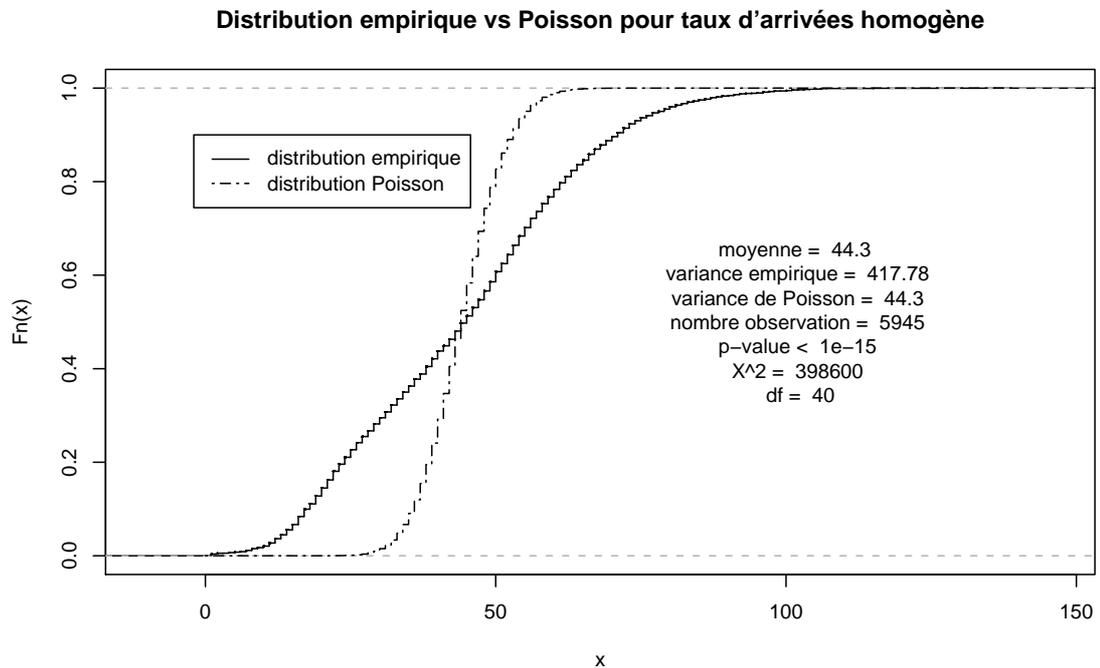


Figure 2.1 – Distribution empirique du nombre moyen d'arrivées par demi-heure vs distribution d'une loi de Poisson avec un taux d'arrivée homogène.

dans l'intervalle k . Chacun des intervalles doit contenir au minimum une valeur espérée de 5 observations, comme il est suggéré dans [34]. Nous calculons X^2 comme suit :

$$X^2 = \sum_{k=0}^m \frac{(O_k - E_k)^2}{E_k}.$$

La valeur obtenue dans notre cas est de $X^2 = 398600$. On peut montrer que la distribution de X^2 est approximativement la distribution d'une chi-deux avec 40 degrés de liberté (sous H_0). Ceci nous permet de calculer une p -valeur qui représente la probabilité que X^2 prenne une valeur supérieure ou égale en supposant que H_0 soit vrai. Dans notre cas, la p -valeur est égale à :

$$p = \Pr(X_{40}^2 > 398600) < 10^{-15}, \quad (2.4)$$

où X_{40}^2 est une variable aléatoire qui suit la loi de chi-deux avec 40 de degrés de liberté. Ainsi, nous devons conclure que cette modélisation n'est pas réaliste et elle doit être rejetée.

Dans le tableau 2.1, on peut voir que le nombre moyen d'arrivées par demi-heure semble être dépendant de l'heure de la journée. Il est important de noter que le tableau 2.1 se base

uniquement sur les journées de la semaine, c'est-à-dire qu'on élimine la journée du samedi qui a trop de particularités (heures d'ouverture différentes, plus petit groupe d'agents) pour être incluse dans la modélisation globale du centre d'appels. De plus, nous avons éliminé 5 autres journées (ex. : veille de Noël, veille du Jour de l'An) pour les mêmes raisons. La variation observée dans le tableau 2.1 sur le nombre moyen d'arrivées nous incite à considérer une autre modélisation assez simple : un processus de Poisson avec un taux variable tout au long de la journée. Le taux d'arrivée serait défini par une fonction $\lambda(t)$ évoluant dans le temps t . Dans notre cas, nous allons avoir une fonction $\lambda(t)$ avec une période correspondant aux 24 heures d'une journée qui sera positive seulement durant les heures d'ouverture du centre. Afin d'obtenir ce type de modélisation, il nous suffit de définir $\hat{\lambda}(t)$ une fonction estimant $\lambda(t)$ à l'aide des données disponibles. Nous allons supposer que $\lambda(t)$ est une fonction constante par demi-heure, i.e., $\lambda(t) = \lambda_i$ lorsque t est dans la demi-heure i . Dans ce cas, le nombre d'arrivées pour la demi-heure i suit une loi de Poisson de paramètre λ_i que nous pouvons estimer par $\hat{\lambda}_i$ le nombre moyen d'arrivées correspondant à la demi-heure i . Ceci revient à appliquer la méthode du maximum de vraisemblance (EVM) pour une loi de Poisson sur chaque demi-heure. Nous allons définir $\hat{\lambda}(t) = \hat{\lambda}_i$ si t est dans la demi-heure i . Nous présentons la fonction $\hat{\lambda}(t)$ dans la figure 2.2 qui prend ses valeurs dans le tableau 2.1. Le tableau 2.1 montre bien qu'on obtient pour chaque demi-heure i un estimateur \hat{s}_i^2 de la variance du nombre d'arrivées (formule (2.3)) beaucoup plus grand que la moyenne du nombre d'arrivées $\hat{\lambda}_i$. Ceci contredit l'hypothèse que les arrivées de la demi-heure i suivent un processus de Poisson stationnaire. Toutefois, pour se convaincre que le modèle n'est pas bon, nous avons appliqué le test statistique de chi-deux sur chaque demi-heure. Le tableau 2.1 présente les p -valeurs p_i pour chaque demi-heure i . Encore une fois, elles sont assez faibles pour conclure que ce processus ne représente pas bien les arrivées d'appels entrants.

Nous n'avons pas obtenu un grand succès avec les deux processus proposés jusqu'ici. La troisième modélisation fera l'hypothèse que le taux d'arrivée dépend non seulement de l'heure de la journée mais aussi du type de la journée : lundi, mardi, mercredi, etc. En d'autres mots, nous allons supposer que nous avons un processus de Poisson avec une fonction de taux constante par morceaux spécifique à chaque jour de la semaine. Cette approche semble être prometteuse car chaque jour de semaine présente un comportement différent. La méthode utilisée pour tester cette modélisation est encore une fois le test du chi-deux. L'ajustement de ce processus ainsi que son test statistique nécessite le regroupement

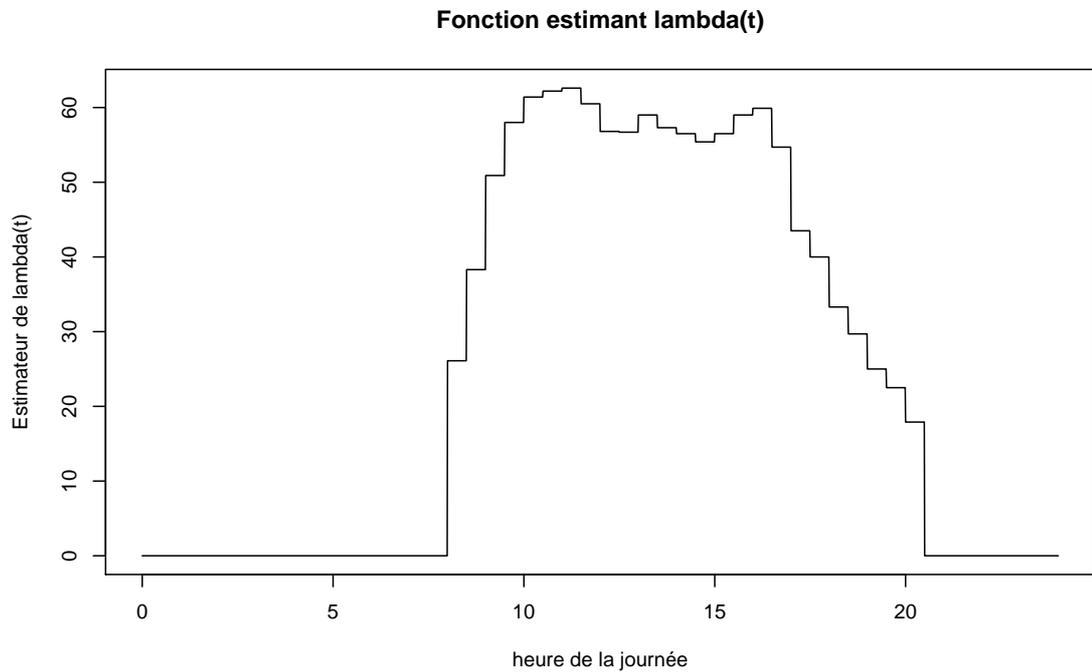


Figure 2.2 – Fonction $\hat{\lambda}(t)$ sur une journée de 24 heures.

des données par demi-heure de chaque type de jour. Les tableaux 2.2 et 2.3 présentent les valeurs obtenues. Ce tableau nous permet de voir que nous n'avions pas tort de séparer les jours de la semaine. Nous observons un trafic plus important pour les lundi et mardi tandis qu'on remarque un faible trafic pour la journée du samedi. Toutefois, les p -valeurs obtenues montrent bien que ce modèle n'est toujours pas réaliste.

Jusqu'à présent, nous avons tenté de trouver une bonne modélisation en ajoutant des dépendances à des variables extérieures. Ces ajouts ont permis de réduire quelque peu la variance observée dans le nombre d'arrivées par 30 minutes. Cependant, nous avons vu que ces améliorations n'ont pas été suffisantes. Un processus de Poisson dont le taux varie selon une fonction déterministe, même si elle dépend de variables extérieures, ne parvient pas à représenter toute la variance dans le nombre d'arrivées par demi-heure. Il existe plusieurs méthodes pour augmenter la variance dans un processus d'arrivée. Parmi celles-ci, on retrouve un type de processus introduit en 1955 par Sir David Cox, appelé *processus de Poisson doublement stochastique* [52]. L'idée derrière ce processus est que la fonction de taux $\{\lambda(t), t \geq 0\}$ est elle-même un processus stochastique.

	lundi		mardi		mercredi	
Demi-heure i	$\hat{\lambda}_i$	p_i	$\hat{\lambda}_i$	p_i	$\hat{\lambda}_i$	p_i
8h00-8h30	31,6	0,33	26,5	0,012	23,2	0,18
8h30-9h00	45,6	0,069	38,4	0,5	36,8	0,074
9h00-9h30	58,3	0,035	52,4	0,036	48,9	0,024
9h30-10h00	66,6	0,38	61,1	0,00024	55	0,057
10h00-10h30	72,5	0,018	63,8	0,066	55,8	0,23
10h30-11h00	74,2	0,0064	62,2	0,013	58,5	0,82
11h00-11h30	72,4	0,0058	66,2	0,083	56	0,071
11h30-12h00	71,1	0,0082	59,8	0,083	57,1	0,31
12h00-12h30	67,8	0,00014	58,8	0,0042	51,3	0,18
12h30-13h00	68,4	0,0055	57,4	0,093	52,5	0,56
13h00-13h30	71,4	2,8e-05	58	6,3e-05	56,5	0,022
13h30-14h00	69,6	4,3e-09	57,8	0,0067	50,3	0,053
14h00-14h30	67,2	0,029	57,3	0,016	51,4	0,01
14h30-15h00	67,7	0,0079	54,8	0,0098	51,3	0,023
15h00-15h30	67,5	0,00042	57,8	0,19	52,9	0,052
15h30-16h00	72,2	0,0066	58,8	0,014	56,4	0,12
16h00-16h30	71,4	0,0044	60,1	0,077	56,1	0,11
16h30-17h00	66,3	2,5e-05	54,3	0,019	52,4	0,23
17h00-17h30	49,8	0,0016	46,3	0,0088	42	0,079
17h30-18h00	48,6	0,00069	40,5	0,024	38,3	0,035
18h00-18h30	39,3	0,057	35	0,0069	33,5	0,037
18h30-19h00	34,4	0,055	31,2	0,15	30,7	0,0078
19h00-19h30	30,3	0,24	27,1	0,027	24,9	0,034
19h30-20h00	27,3	0,36	24,7	0,016	22,4	0,081
20h00-20h30	19,7	0,12	17,9	0,00021	19,4	0,034

Tableau 2.2 – Nombre moyen d’arrivées par demi-heure pour les appels entrants selon le type de jour. Nous donnons aussi la p -valeur associée à chaque demi-heure de chaque jour en supposant que nous avons un processus de Poisson de taux constant. Nous considérons les jours de lundi, mardi et mercredi.

	jeudi		vendredi		samedi	
Demi-heure i	$\hat{\lambda}_i$	p_i	$\hat{\lambda}_i$	p_i	$\hat{\lambda}_i$	p_i
8h00-8h30	24,7	0,39	25,1	0,16	-	-
8h30-9h00	36	0,59	36	0,011	16	0,048
9h00-9h30	49,2	0,039	47	0,62	18,6	0,077
9h30-10h00	54,3	0,13	54,5	0,43	22,3	0,029
10h00-10h30	60,2	0,026	56,3	0,00067	24	0,0053
10h30-11h00	57,8	0,32	60	0,0036	27,3	0,00012
11h00-11h30	60,5	0,01	59,5	0,016	25,1	0,11
11h30-12h00	57,3	0,02	58,4	1,4e-05	24,7	0,16
12h00-12h30	54,3	0,019	53,1	1,1e-06	24,9	0,032
12h30-13h00	54,7	0,0037	51,9	0,02	22,9	0,65
13h00-13h30	56,8	0,19	53,7	8e-05	21	0,84
13h30-14h00	54,7	0,019	55,8	0,00029	21,7	0,0028
14h00-14h30	54,2	0,23	53,9	0,042	19,2	0,44
14h30-15h00	51,6	0,0059	53	6,2e-05	18,6	0,51
15h00-15h30	50,5	0,43	55,5	0,045	17,3	0,14
15h30-16h00	55,3	0,0096	54,2	0,012	16,8	0,71
16h00-16h30	58,9	0,00029	54,4	0,0039	17	0,38
16h30-17h00	52,6	0,00051	49,3	0,0038	14,3	0,38
17h00-17h30	41,7	0,04	38,9	0,01	-	-
17h30-18h00	38,7	0,2	35,1	0,054	-	-
18h00-18h30	32,6	0,0061	27,5	0,055	-	-
18h30-19h00	29,8	0,13	23,3	0,52	-	-
19h00-19h30	24,1	0,002	19,6	0,059	-	-
19h30-20h00	23	0,95	15,9	0,52	-	-
20h00-20h30	18,8	0,24	14,4	0,46	-	-

Tableau 2.3 – Nombre moyen d’arrivées par demi-heure pour les appels entrants selon le type de jour. Nous donnons aussi la p -valeur associée à chaque demi-heure de chaque jour en supposant que nous avons un processus de Poisson de taux constant. Nous considérons les jours de jeudi, vendredi et samedi.

Cette approche a été utilisée par Jongbloed et Koole [31] avec le même type de données que nous disposons, i.e., le nombre d'arrivées agrégées par demi-heure. Suite aux bons résultats qu'ils ont obtenus, nous avons expérimenté leur méthode sur notre cas. Dans [31], on suppose que les demi-heures sont indépendantes entre elles et que les arrivées d'appels entrants suivent un processus de Poisson doublement stochastique spécifique à chaque demi-heure. Plus précisément, on y fait la supposition que le taux d'arrivée Λ_i , pour la i -ième demi-heure, est une variable aléatoire de densité h_i définie sur l'intervalle $[0, \infty)$. Une fois que le taux est fixé pour une demi-heure, il reste constant pour les 30 prochaines minutes. En d'autres mots, si Λ_i prend la valeur λ_i alors nous savons que les arrivées vont se faire selon un processus de Poisson homogène de taux λ_i jusqu'à la prochaine demi-heure.

Sous l'hypothèse d'avoir un tel processus, la connaissance du paramètre λ_i implique que le nombre X_i d'arrivées d'appels durant la i -ième demi-heure suit une loi de Poisson de paramètre λ_i . Cette remarque nous permet de définir la probabilité inconditionnelle d'avoir x_i arrivées durant la i -ième demi-heure :

$$P_i(X_i = x_i) = \int_0^\infty \frac{\lambda_i^{x_i}}{x_i!} e^{-\lambda_i} h_i(\lambda_i) d\lambda_i. \quad (2.5)$$

Afin d'utiliser ce processus, il suffit de préciser la forme de h_i pour chaque demi-heure i .

Dans ce qui suit, nous allons expliquer le traitement pour une demi-heure i en particulier. Ceci est permis car nous avons supposé que les demi-heures sont indépendantes entre elles.

Le choix de la fonction de densité h_i est loin d'être évident. Il existe toutefois une famille de fonctions attrayante, car assez flexible et simple d'utilisation. Cette famille est l'ensemble des fonctions de densité h_{α_i, β_i} d'une loi gamma de paramètres α_i et β_i pour la demi-heure i :

$$h_{\alpha_i, \beta_i}(\lambda) = \frac{\beta_i^{-\alpha_i}}{\Gamma(\alpha_i)} \lambda^{\alpha_i-1} e^{-\lambda/\beta_i} I_{[0, \infty)}(\lambda), \quad (2.6)$$

où

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

Ce choix de fonction implique que l'équation (2.5) est maintenant définie par :

$$\begin{aligned} P_{\alpha_i, \beta_i}(X_i = x_i) &= \int_0^\infty \frac{\lambda^{x_i}}{x_i!} e^{-\lambda} h_{\alpha_i, \beta_i}(\lambda) d\lambda \\ &= \frac{\beta_i^{-\alpha_i}}{x_i! \Gamma(\alpha_i)} \int_0^\infty \lambda^{x_i + \alpha_i - 1} e^{-\lambda(1/\beta_i + 1)} d\lambda \\ &= \frac{\Gamma(x_i + \alpha_i)}{x_i! \Gamma(\alpha_i)} \left(\frac{1}{1 + \beta_i} \right)^{\alpha_i} \left(1 - \frac{1}{1 + \beta_i} \right)^{x_i} \end{aligned} \quad (2.7)$$

$$= \binom{\alpha_i + x_i - 1}{x_i} \left(\frac{1}{1 + \beta_i} \right)^{\alpha_i} \left(1 - \frac{1}{1 + \beta_i} \right)^{x_i}, \quad \alpha_i \in \mathbb{N}. \quad (2.8)$$

Les équations (2.7) et (2.8) correspondent à la fonction de masse d'une loi binomiale négative avec un α_i réel dans le premier cas et un α_i entier dans le deuxième. Ainsi, ce développement montre que X_i suit une loi binomiale négative. Cette loi correspond au nombre d'échecs avant d'obtenir α_i succès dans une expérience de Bernoulli où la probabilité de succès est $\varphi_i = 1/(1 + \beta_i)$. Ainsi, l'estimation des paramètres de la loi h_{α_i, β_i} se ramène à l'estimation des paramètres d'une loi binomiale négative.

Pour estimer ces paramètres à partir des données, nous allons utiliser la méthode du maximum de vraisemblance de la loi binomiale négative. Les estimateurs $\hat{\alpha}_i$ et $\hat{\varphi}_i$ sont ceux qui maximisent la fonction de vraisemblance :

$$\phi_i(\alpha_i, \varphi_i) = \prod_{j=1}^r g_{\alpha_i, \varphi_i}(x_{i,j}), \quad (2.9)$$

où g_{α_i, φ_i} est la fonction de masse de la loi binomiale négative de la demi-heure i , r le nombre de journées et les $x_{i,j}$ sont le nombre d'arrivées pour la demi-heure i de la journée j . On suppose que les $x_{i,1}, \dots, x_{i,r}$ sont i.i.d. suivant la fonction de masse g_{α_i, φ_i} . Dans certain cas, il est plus pratique de maximiser le logarithme naturel de cette fonction au lieu de la fonction elle-même [45]. Dans [31], les auteurs proposent une fonction de log-vraisemblance à maximiser qui dépend uniquement du paramètre α_i :

$$\tilde{\phi}_i'(\alpha_i) = \frac{1}{r} \sum_{j=1}^r \sum_{k=1}^{x_{i,j}} \ln(\alpha_i + k - 1) - (\alpha_i + \bar{x}_i) \ln(\alpha_i + \bar{x}_i), \quad (2.10)$$

où \bar{x}_i est le nombre moyen d'arrivées pour la demi-heure i . Cette fonction à maximiser n'est pas la seule qui permet de trouver l'EVM de α_i . On retrouve dans [37] une autre fonction avec le même maximum :

$$\tilde{\phi}_i''(\alpha_i) = \sum_{k=1}^{m_i} f_i(k) \ln(\alpha_i + k - 1) + r\alpha_i \ln(\alpha_i/(\alpha_i + \bar{x}_i)) + r\bar{x}_i \ln(\bar{x}_i/(\alpha_i + \bar{x}_i)), \quad (2.11)$$

où $m_i = \max_{1 \leq j \leq r} x_{i,j}$ et $f_i(k)$ est le nombre de valeurs de j telles que $x_{i,j} \geq k$. L'avantage de cette deuxième fonction est que l'on peut calculer son maximum plus rapidement.

On peut facilement obtenir le maximum $\hat{\alpha}_i$ de ces deux fonctions en utilisant la méthode de Newton. Une fois que nous avons calculé $\hat{\alpha}_i$, on obtient l'estimateur $\hat{\varphi}_i$ en utilisant la formule : $\hat{\varphi}_i = \hat{\alpha}_i/(\hat{\alpha}_i + \bar{x}_i)$. Finalement, on peut calculer l'estimateur du paramètre β_i de la loi gamma par $\hat{\beta}_i = (1 - \hat{\varphi}_i)/\hat{\varphi}_i$.

Après avoir calculé les estimateurs, la prochaine étape est de tester l'hypothèse selon laquelle cette modélisation représente bien la réalité. Pour cela, nous avons utilisé le test statistique de Kolmogorov-Smirnov avec "bootstrap" sur chaque demi-heure i . Ce test est caractérisé par la mesure :

$$D_i = \sup_{x>0} |\hat{F}_i(x) - F_{\hat{\alpha}_i, \hat{\beta}_i}(x)|, \quad (2.12)$$

où \hat{F}_i est la fonction de répartition empirique du nombre d'arrivées pour la demi-heure i et $F_{\hat{\alpha}_i, \hat{\beta}_i}$ est la fonction de répartition de la loi binomiale négative ayant les paramètres obtenus par la méthode EVM. En fait, cette mesure donne l'écart maximal entre la distribution observée et celle qui fut estimée. Une valeur suffisamment grande pour ce test statistique contredirait l'hypothèse nulle :

$$H_0 \quad : \quad h_i \equiv h_{\alpha_i, \beta_i} \text{ pour } \alpha_i, \beta_i \geq 0, \quad (2.13)$$

Nous ne pouvons pas appliquer directement le test de Kolmogorov-Smirnov parce que les paramètres α_i et β_i ont été estimés avec les mêmes données que pour calculer \hat{F}_i . C'est pour cette raison que nous allons utiliser la méthode du "bootstrap" pour approximer la distribution de la mesure D_i sous l'hypothèse nulle.

Voici la méthode pour obtenir une observation de la distribution "bootstrap" de D_i . Premièrement, on doit générer, à partir de la répartition $F_{\hat{\alpha}_i, \hat{\beta}_i}$, un échantillon de taille équivalente au nombre de journées observées. Avec cet échantillon, on estime de nouveaux paramètres $\hat{\alpha}_i^*$ et $\hat{\beta}_i^*$ à l'aide de l'EVM, et on calcule la distance maximale d_i^* entre la fonction de répartition empirique de cet échantillon et la fonction de répartition $F_{\hat{\alpha}_i^*, \hat{\beta}_i^*}$. En répétant ce processus un grand nombre de fois (disons B répétitions), l'ensemble des distances maximales calculées, $d_{i,1}^*, d_{i,2}^*, \dots, d_{i,B}^*$ représente une approximation de la distribution de D_i sous l'hypothèse nulle. En définissant par d_i la valeur de D_i obtenue avec les données originales, on peut calculer une approximation de la p -valeur pour la demi-heure i :

$$p_i \approx \frac{\text{Card}\{1 \leq k \leq B : d_{i,k}^* \geq d_i\}}{B}, \quad (2.14)$$

où Card définit la cardinalité d'un ensemble.

Nous avons utilisé $B = 1000$ échantillons pour obtenir une bonne approximation de la p -valeur sur chaque demi-heure. Le tableau 2.4 présente les résultats obtenus.

Les résultats montrent bien que cette modélisation doit être rejetée dans le cas où nous supposons qu'elle permet de modéliser l'ensemble de toutes les journées (première séparation

Demi-heure i	Tous les jours			Jours de semaine		
	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i
8h00-8h30	14,6	1,8	0,20	14,6	1,8	0,20
8h30-9h00	8,2	4,2	ϵ	33,0	1,2	0,13
9h00-9h30	6,7	6,8	ϵ	26,4	1,9	0,35
9h30-10h00	7,4	7,0	ϵ	30,6	1,9	0,68
10h00-10h30	7,3	7,5	ϵ	23,8	2,6	0,81
10h30-11h00	9,0	6,2	ϵ	30,0	2,1	0,37
11h00-11h30	7,3	7,6	ϵ	25,0	2,5	0,59
11h30-12h00	7,4	7,3	ϵ	22,0	2,7	0,78
12h00-12h30	8,0	6,4	ϵ	20,8	2,7	0,80
12h30-13h00	7,5	6,8	ϵ	22,4	2,5	0,57
13h00-13h30	6,0	8,7	ϵ	17,2	3,4	0,58
13h30-14h00	6,2	8,2	ϵ	17,8	3,2	0,31
14h00-14h30	5,7	8,8	ϵ	21,4	2,6	0,62
14h30-15h00	5,8	8,4	ϵ	19,6	2,8	0,94
15h00-15h30	5,6	8,8	ϵ	26,1	2,2	0,17
15h30-16h00	5,1	10,2	ϵ	24,7	2,4	0,98
16h00-16h30	5,1	10,2	ϵ	26,9	2,2	0,93
16h30-17h00	4,5	10,6	ϵ	22,6	2,4	0,24
17h00-17h30	15,2	2,9	0,11	15,2	2,9	0,11
17h30-18h00	16,7	2,4	0,04	16,7	2,4	0,04
18h00-18h30	13,8	2,4	0,53	13,8	2,4	0,53
18h30-19h00	11,9	2,5	0,26	11,9	2,5	0,26
19h00-19h30	9,3	2,7	ϵ	9,3	2,7	ϵ
19h30-20h00	12,4	1,8	0,10	12,4	1,8	0,10
20h00-20h30	11,1	1,6	0,31	11,1	1,6	0,31

Tableau 2.4 – Estimation des paramètres α_i et β_i pour chaque demi-heure i . Nous donnons la p -valeur associée à ces paramètres sous l’hypothèse que nous avons un processus de Poisson doublement stochastique avec une loi gamma.

du tableau). Nous avons des approximations de la p -valeur acceptables (supérieurs à 0,15) pour quelques demi-heures, mais nous observons trop souvent des approximations de la p -valeur égale à ϵ . La valeur ϵ signifie que nous n'avons pas observé de $d_{i,k}^*$ supérieure à d_i . Ainsi nous pouvons conclure que la vraie p -valeur doit être très petite, i.e., p -valeur $< 0,001$. Il est intéressant de noter que toutes les p -valeurs "raisonnables" sont associées à des demi-heures où le centre est fermé durant la journée du samedi. Ceci nous laisse croire que le jour de la semaine pourrait avoir une influence sur le processus doublement stochastique. Nous avons fait le même exercice en éliminant les jours du samedi dans les données. Les résultats se trouvent dans la deuxième séparation du tableau 2.4. Maintenant, nous ne pouvons pas rejeter statistiquement ce processus. Nous espérons obtenir ces bons résultats, car on n'avait pas réussi à rejeter cette modélisation dans les rapports [7, 8] avec moins de données disponibles. De plus, il est important de mentionner que ces rapports analysaient seulement des données recueillies à partir de 14h et, que la modélisation semble bien s'ajuster pour les demi-heures avant 14h.

Dans le prochain modèle, nous faisons l'hypothèse additionnelle que chaque demi-heure d'un jour de la semaine (lundi, mardi,...) peut être modélisée par ce type de processus doublement stochastique. Il pourrait être intéressant d'utiliser cette méthode dans le cas où nous voudrions analyser une journée en particulier. Nous avons calculé les paramètres $\hat{\alpha}_i$ et $\hat{\beta}_i$ ainsi qu'une approximation p_i de la p -valeur pour les processus d'une demi-heure de chaque jour. Les tableaux 2.5 et 2.6 présentent les résultats obtenus.

Les approximations de la p -valeur obtenues dans les tableaux 2.5 et 2.6 sont encore une fois assez bonnes. Bien que ces modélisations semblent assez robustes pour représenter les arrivées d'appels entrants, l'hypothèse d'avoir une indépendance entre les demi-heures successives est un peu forte à notre goût. Intuitivement, si nous remarquons un taux d'arrivée élevé durant les 30 dernières minutes alors nous pouvons nous attendre à observer un taux d'arrivée élevé pour la prochaine demi-heure. Les tests statistiques utilisés ne rejetaient pas le processus sur chaque demi-heure, mais ils ne testaient pas l'indépendance des demi-heures. Nous avons calculé les corrélations empiriques entre le nombre d'arrivées de chaque demi-heure et les valeurs obtenues sont près de 0,5. Les p -valeurs d'un test statistique de MATLAB [43] montrent que les corrélations sont significatives. Ainsi, il pourrait être intéressant d'étudier un processus avec une certaine dépendance entre les demi-heures. Il existe plusieurs manières d'introduire de la corrélation entre les taux d'arrivée de demi-

	lundi			mardi			mercredi		
Demi-heure i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i
8h00-8h30	16,9	1,9	0,07	19,0	1,4	0,30	8,6	2,7	ϵ
8h30-9h00	38,3	1,2	ϵ	62,0	0,6	0,70	60,2	0,6	0,10
9h00-9h30	13,6	4,3	ϵ	24,6	2,1	0,74	40,1	1,2	0,95
9h30-10h00	26,6	2,5	0,05	31,4	1,9	0,14	58,2	0,9	0,94
10h00-10h30	21,6	3,4	0,11	30,7	2,1	0,20	123,4	0,5	0,77
10h30-11h00	34,7	2,1	0,58	35,6	1,7	0,77	94,0	0,6	0,74
11h00-11h30	35,3	2,1	0,53	34,0	1,9	0,84	29,6	1,9	0,26
11h30-12h00	23,8	3,0	0,94	24,2	2,5	0,80	30,3	1,9	0,14
12h00-12h30	24,4	2,8	0,86	30,8	1,9	0,59	23,6	2,2	0,12
12h30-13h00	24,3	2,8	0,05	13,3	4,3	ϵ	67,8	0,8	0,45
13h00-13h30	15,9	4,5	0,30	15,2	3,8	0,40	36,4	1,6	0,99
13h30-14h00	17,1	4,1	0,42	26,4	2,2	0,36	17,3	2,9	0,24
14h00-14h30	27,4	2,5	0,55	38,2	1,5	0,25	13,2	3,9	0,20
14h30-15h00	18,3	3,7	0,46	23,1	2,4	0,64	25,8	2,0	0,01
15h00-15h30	18,5	3,7	0,99	73,5	0,8	0,58	35,8	1,5	0,51
15h30-16h00	22,1	3,3	0,64	28,8	2,0	0,70	52,5	1,1	0,91
16h00-16h30	24,8	2,9	0,64	64,2	0,9	0,09	59,7	0,9	0,63
16h30-17h00	24,5	2,7	0,08	69,6	0,8	0,11	22,7	2,3	0,10
17h00-17h30	16,5	3,0	0,23	28,4	1,6	0,68	7,1	6,0	ϵ
17h30-18h00	28,5	1,7	0,63	28,3	1,4	0,59	7,8	4,9	0,01
18h00-18h30	19,5	2,0	0,28	23,9	1,5	0,04	16,9	2,0	0,37
18h30-19h00	26,9	1,3	0,26	9,5	3,3	0,05	16,2	1,9	0,91
19h00-19h30	11,0	2,8	0,01	14,7	1,8	0,56	16,3	1,5	0,37
19h30-20h00	31,4	0,9	0,09	23,0	1,1	0,20	18,1	1,2	0,23
20h00-20h30	51,7	0,4	0,54	5,8	3,1	0,16	22,4	0,9	0,91

Tableau 2.5 – Estimation des paramètres α_i et β_i pour chaque demi-heure i des jours du lundi, mardi et mercredi. Nous donnons l'approximation de la p -valeur associée à ces paramètres sous l'hypothèse que nous avons un processus de Poisson doublement stochastique avec une loi gamma.

	jeudi			vendredi			samedi		
Demi-heure i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i
8h00-8h30	48,0	0,5	0,90	21,3	1,1	0,96	—	—	—
8h30-9h00	40,1	0,8	0,52	27,1	1,3	0,55	14,7	1,0	0,46
9h00-9h30	56,1	0,9	0,70	36,8	1,2	0,03	20,8	0,9	0,92
9h30-10h00	13,3	3,7	0,04	24,7	2,1	0,12	18,9	1,2	0,81
10h00-10h30	6,6	8,2	0,001	13,0	4,0	0,23	28,8	0,8	0,57
10h30-11h00	39,7	1,4	0,28	13,6	4,1	ϵ	32,1	0,8	0,43
11h00-11h30	7,1	7,5	0,001	15,3	3,6	0,001	16,5	1,5	0,61
11h30-12h00	5,8	8,7	ϵ	6,5	8,2	ϵ	23,8	1,0	0,50
12h00-12h30	7,9	6,2	ϵ	10,3	4,6	0,005	17,9	1,3	0,20
12h30-13h00	4,6	10,2	ϵ	33,9	1,5	0,44	27,6	0,8	0,79
13h00-13h30	15,3	3,5	0,05	20,1	2,6	0,31	52,5	0,4	0,27
13h30-14h00	43,0	1,3	0,91	20,5	2,7	0,61	14,3	1,5	0,88
14h00-14h30	37,8	1,4	0,56	26,4	2,0	0,40	12,9	1,5	0,34
14h30-15h00	25,1	2,1	0,94	35,6	1,4	0,62	51,7	0,4	0,42
15h00-15h30	51,6	1,0	0,11	27,9	2,0	0,70	62,1	0,3	0,82
15h30-16h00	40,0	1,4	0,34	42,5	1,3	0,73	42,7	0,4	0,97
16h00-16h30	37,4	1,6	0,89	27,7	1,9	0,89	40,4	0,4	0,61
16h30-17h00	22,3	2,3	0,38	44,6	1,1	0,55	20,7	0,7	0,66
17h00-17h30	26,9	1,6	0,51	24,1	1,6	0,66	—	—	—
17h30-18h00	47,3	0,8	0,39	22,5	1,6	0,86	—	—	—
18h00-18h30	23,3	1,4	0,74	16,5	1,6	0,10	—	—	—
18h30-19h00	11,2	2,6	0,11	12,1	1,9	0,07	—	—	—
19h00-19h30	11,0	2,2	0,003	7,5	2,6	0,005	—	—	—
19h30-20h00	16,6	1,4	0,16	18,9	0,8	0,20	—	—	—
20h00-20h30	13,5	1,4	0,32	19,2	0,7	0,58	—	—	—

Tableau 2.6 – Estimation des paramètres de α_i et β_i pour chaque demi-heure i des jours du jeudi, vendredi et samedi. Nous donnons l’approximation de la p -valeur associée à ces paramètres sous l’hypothèse que nous avons un processus de Poisson doublement stochastique avec une loi gamma.

heures successives. Une façon fort simple est de définir pour *chaque journée* un facteur qui multipliera une fonction de taux d'arrivée déterministe. On fera l'hypothèse que ce facteur est une variable aléatoire suivant une fonction de répartition F .

L'idée d'appliquer ce genre de processus nous est venue suite à la lecture d'un article de Whitt [56] qui propose l'utilisation d'un taux d'arrivée stochastique $\Lambda(t)$ de la forme :

$$\Lambda(t) = W\lambda(t), t \geq 0, \quad (2.15)$$

où $\lambda(t)$ est une fonction représentant la variation du taux d'arrivée selon le temps t et W est une variable aléatoire positive et indépendante des processus dans le centre d'appels. La variable W est valable pour une journée. L'idée derrière ce modèle est d'augmenter la variance dans le nombre d'arrivées.

Pour ce processus, nous allons supposer que $\lambda(t) = \lambda_i$ lorsque t est dans la demi-heure i et que W suit une loi gamma de paramètres α' et β' . La fonction de densité $h_{\alpha',\beta'}$ de W est décrite par l'équation (2.6). Cette loi semble un bon choix car elle est définie positivement et permet une bonne flexibilité.

La prochaine étape est de trouver une méthode pour estimer les paramètres de ce processus avec les données disponibles. Il nous semble que la meilleure méthode est d'utiliser les valeurs qui maximisent sa fonction de vraisemblance. Nous allons maintenant décrire la démarche pour estimer les paramètres du processus.

Soit $X_{i,j}$ le nombre d'arrivées dans la demi-heure i de la j -ième journée, nous voulons obtenir la fonction de masse conjointe de ces variables aléatoires avec $i \in \{1, \dots, 25\}$ et $j \in \{1, \dots, r\}$. Nous allons dénoter cette fonction par $f_{\theta}(x_{1,1}, x_{2,1}, \dots, x_{25,1}, x_{1,2}, \dots, x_{25,r})$ où $\theta = (\alpha', \beta', \lambda_1, \dots, \lambda_{25})$ est l'ensemble des paramètres du processus. En supposant que nous observons des journées indépendantes, nous pouvons écrire la fonction de masse ou de vraisemblance :

$$f_{\theta}(x_{1,1}, x_{2,1}, \dots, x_{25,1}, x_{1,2}, \dots, x_{25,r}) = \prod_{j=1}^r \tilde{f}_{\theta}(x_{1,j}, x_{2,j}, \dots, x_{25,j}), \quad (2.16)$$

où $\tilde{f}_{\theta}(x_{1,j}, x_{2,j}, \dots, x_{25,j})$ est la fonction de masse conjointe pour le nombre d'arrivées par demi-heure d'un jour j .

Nous allons maintenant définir la fonction de masse $\tilde{f}_{\theta}(x_{1,j}, x_{2,j}, \dots, x_{25,j})$ pour la journée j . En conditionnant sur la valeur de $W = w$, nous obtenons un processus de Poisson avec

une fonction de taux $\Lambda(t) = w\lambda_i$ lorsque t est dans la demi-heure i . En fixant $j = 1$, nous obtenons :

$$\begin{aligned}
\tilde{f}_\theta(x_{1,1}, x_{2,1}, \dots, x_{25,1}) &= \int_0^\infty \tilde{f}_\theta(x_{1,1}, x_{2,1}, \dots, x_{25,1}|w) h_{\alpha', \beta'}(w) dw \\
&= \int_0^\infty \prod_{i=1}^{25} \left(\frac{(\lambda_i w)^{x_{i,1}} e^{-\lambda_i w}}{x_{i,1}!} \right) \left(\frac{\beta'^{-\alpha'} w^{\alpha'-1} e^{-w/\beta'}}{\Gamma(\alpha')} \right) dw \\
&= \frac{\beta'^{-\alpha'} \left(\prod_{i=1}^{25} \lambda_i^{x_{i,1}} \right)}{\Gamma(\alpha') \left(\prod_{i=1}^{25} x_{i,1}! \right)} \int_0^\infty w^{\sum_{i=1}^{25} x_{i,1} + \alpha' - 1} e^{-w(\sum_{i=1}^{25} \lambda_i + 1/\beta')} dw \\
&= \frac{\beta'^{-\alpha'} \left(\prod_{i=1}^{25} \lambda_i^{x_{i,1}} \right) \Gamma(\sum_{i=1}^{25} x_{i,1} + \alpha')}{\Gamma(\alpha') \left(\prod_{i=1}^{25} x_{i,1}! \right) \left(\sum_{i=1}^{25} \lambda_i + 1/\beta' \right)^{\sum_{i=1}^{25} x_{i,1} + \alpha'}} \\
&= \frac{\Gamma(\sum_{i=1}^{25} x_{i,1} + \alpha')}{\Gamma(\alpha') \left(\prod_{i=1}^{25} x_{i,1}! \right)} \left(\frac{1}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right)^{\alpha'} \prod_{i=1}^{25} \left(\frac{\beta' \lambda_i}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right)^{x_{i,1}}.
\end{aligned}$$

Dans ce résultat, on reconnaît la fonction de masse d'une loi binomiale négative multivariée (multinomiale négative) de paramètres α' et $P_k = \beta' \lambda_k$, $1 \leq k \leq 25$ [30].

La fonction de log-vraisemblance pour ce processus est alors :

$$\begin{aligned}
\phi(\alpha', \beta', \lambda_1, \dots, \lambda_{25}) &= \sum_{j=1}^r \log(\tilde{f}_\theta(x_{1,j}, x_{2,j}, \dots, x_{25,j})) \\
&= \sum_{j=1}^r \left(\log \left(\frac{\Gamma(\sum_{i=1}^{25} x_{i,j} + \alpha')}{\Gamma(\alpha')} \right) + \Delta + \alpha' \log \left(\frac{1}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \right. \\
&\quad \left. + \sum_{i=1}^{25} x_{i,j} \log \left(\frac{\beta' \lambda_i}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \right) \\
&= \sum_{j=1}^r \left(\sum_{l=1}^{\sum_{i=1}^{25} x_{i,j}} \log(\alpha' + l - 1) + \Delta + \alpha' \log \left(\frac{1}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \right. \\
&\quad \left. + \sum_{i=1}^{25} x_{i,j} \log \left(\frac{\beta' \lambda_i}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \right) \\
&= \sum_{l=1}^q o_l \log(\alpha' + l - 1) + \Delta + r\alpha' \log \left(\frac{1}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \\
&\quad + \sum_{j=1}^r \left(\sum_{i=1}^{25} x_{i,j} \log \left(\frac{\beta' \lambda_i}{\sum_{k=1}^{25} \beta' \lambda_k + 1} \right) \right), \tag{2.17}
\end{aligned}$$

où Δ est une constante indépendante des paramètres, $q = \max_{1 \leq j \leq r} \sum_{i=1}^{25} x_{i,j}$ et o_l est le nombre de valeurs de j telles que $\sum_{i=1}^{25} x_{i,j} \geq l$. Les fonctions (2.16) et (2.17) atteignent leur maximum avec les mêmes valeurs. En observant ces deux fonctions, on remarque que nous avons un paramètre redondant. Pour n'importe quelle valeur de β' , le maximum peut

être atteint grâce au degré de liberté de chaque λ_i pour $i = 1, \dots, 25$. Ceci nous permet de fixer $\beta' = 1/\alpha'$. Ce choix se justifie par le fait que nous allons ainsi obtenir $E[W] = 1$. Cette remarque permet de définir la fonction :

$$\begin{aligned} \tilde{\phi}(\alpha', \lambda_1, \dots, \lambda_{25}) &= \phi(\alpha', 1/\alpha', \lambda_1, \dots, \lambda_{25}) \\ &= \sum_{l=1}^q o_l \log(\alpha' + l - 1) + \Delta + r\alpha' \log\left(\frac{\alpha'}{\sum_{k=1}^{25} \lambda_k + \alpha'}\right) \\ &\quad + \sum_{j=1}^r \left(\sum_{i=1}^{25} x_{i,j} \log\left(\frac{\lambda_i}{\sum_{k=1}^{25} \lambda_k + \alpha'}\right) \right), \end{aligned} \quad (2.18)$$

dont les points maximaux sont des points maximaux de la fonction (2.17).

Pour maximiser la fonction (2.18), nous avons utilisé le logiciel MATLAB [43] qui offre des méthodes pour maximiser des fonctions à plusieurs paramètres. Afin d'accélérer la maximisation, nous lui avons fourni les dérivées partielles par rapport aux 26 paramètres :

$$\begin{aligned} \frac{\partial \tilde{\phi}(\bullet)}{\partial \alpha'} &= \sum_{l=1}^q o_l (\alpha' + l - 1)^{-1} + r \log(r\alpha') + r \\ &\quad - r \log\left(\sum_{k=1}^{25} \lambda_k + \alpha'\right) - \frac{\tilde{x} + \alpha'}{\sum_{k=1}^{25} \lambda_k + \alpha'} \end{aligned} \quad (2.19)$$

$$\frac{\partial \tilde{\phi}(\bullet)}{\partial \lambda_l} = \frac{\sum_{j=1}^r x_{l,j}}{\lambda_l} - \frac{\tilde{x} + \alpha'}{\sum_{k=1}^{25} \lambda_k + \alpha'} \quad 1 \leq l \leq 25, \quad (2.20)$$

où $\tilde{x} = \sum_{j=1}^r \sum_{i=1}^{25} x_{i,j}$. Une condition nécessaire à un point maximal est que les dérivées partielles (2.19) et (2.20) soient égales à zéro en ce point.

Le tableau 2.7 donne la valeur des paramètres qui maximise la fonction (2.18) avec les données provenant de notre centre d'appels. Le tableau présente un intervalle de confiance de 90% pour chaque paramètre. Cet intervalle a été calculé à l'aide de la méthode "bootstrap" du percentile [45].

Nous allons maintenant étudier de plus près le processus. Nous allons nous intéresser à la variabilité et à la corrélation du nombre d'arrivées par demi-heure. Dans un premier temps, il est possible d'obtenir le coefficient de variation CV du nombre d'arrivées X_i dans la demi-heure i :

$$\begin{aligned} \text{CV}[X_i] &= (\text{VAR}[X_i])^{1/2} / E[X_i] \\ &= (E[\text{VAR}(X_i|W)] + \text{VAR}[E(X_i|W)])^{1/2} / E[E(X_i|W)] \\ &= (E[\lambda_i W] + \text{VAR}[\lambda_i W])^{1/2} / E[\lambda_i W] \end{aligned}$$

Estimateur	Valeur	Intervalle de confiance 90%
$\hat{\alpha}'$	28,93	(24,62,33,97)
$\hat{\beta}'$	0,0346	(0,0292,0,0402)
$\hat{\lambda}_1$	26,1	(25,2,26,9)
$\hat{\lambda}_2$	38,5	(37,3,39,7)
$\hat{\lambda}_3$	51,1	(49,8,52,3)
$\hat{\lambda}_4$	58,2	(56,7,59,9)
$\hat{\lambda}_5$	61,3	(59,3,63,0)
$\hat{\lambda}_6$	61,6	(60,0,63,4)
$\hat{\lambda}_7$	62,2	(60,3,64,1)
$\hat{\lambda}_8$	60,2	(58,3,61,8)
$\hat{\lambda}_9$	56,6	(54,7,58,6)
$\hat{\lambda}_{10}$	56,4	(54,6,57,7)
$\hat{\lambda}_{11}$	58,6	(56,6,60,1)
$\hat{\lambda}_{12}$	57,0	(55,1,58,7)
$\hat{\lambda}_{13}$	56,2	(54,5,57,8)
$\hat{\lambda}_{14}$	55,2	(53,3,56,8)
$\hat{\lambda}_{15}$	56,3	(54,8,57,7)
$\hat{\lambda}_{16}$	58,9	(56,8,60,9)
$\hat{\lambda}_{17}$	59,6	(57,7,61,3)
$\hat{\lambda}_{18}$	54,5	(52,6,55,8)
$\hat{\lambda}_{19}$	43,5	(42,1,44,7)
$\hat{\lambda}_{20}$	40,0	(38,9,41,1)
$\hat{\lambda}_{21}$	33,5	(32,1,34,7)
$\hat{\lambda}_{22}$	29,8	(28,6,31,1)
$\hat{\lambda}_{23}$	25,3	(24,2,26,4)
$\hat{\lambda}_{24}$	22,5	(21,4,23,5)
$\hat{\lambda}_{25}$	18,0	(17,0,18,8)

Tableau 2.7 – Estimation des paramètres α' , β' et λ_i pour un processus d'arrivée de Poisson corrélé par un facteur de loi gamma. Nous donnons un intervalle de confiance de 90% pour chaque estimateur.

$$\begin{aligned}
&= (\lambda_i \alpha' \beta' + \lambda_i^2 \alpha' \beta'^2)^{1/2} / (\lambda_i \alpha' \beta') \\
&= \left(\frac{1 + \lambda_i \beta'}{\lambda_i \beta' \alpha'} \right)^{1/2}.
\end{aligned} \tag{2.21}$$

Nous pouvons supposer que $\lambda_i \beta' \alpha'$ est assez constant lors de l'ajustement du processus, car ce terme doit être près de $E[X_i]$. Ceci implique que $CV[X_i]$ est contrôlé par $\lambda_i \beta'$: plus sa valeur est faible, plus le coefficient de variation est faible et vice versa.

Ensuite, nous avons la corrélation $CORR[X_i, X_j]$ entre le nombre d'arrivées X_i et X_j des demi-heures i et j :

$$\begin{aligned}
CORR[X_i, X_j] &= COV[X_i, X_j] / (\text{VAR}[X_i] \text{VAR}[X_j])^{1/2} \\
&= (\lambda_i \lambda_j \text{VAR}[W]) / \left((\lambda_i \alpha' \beta' + \lambda_i^2 \alpha' \beta'^2) (\lambda_j \alpha' \beta' + \lambda_j^2 \alpha' \beta'^2) \right)^{1/2} \\
&= \left(\frac{\lambda_i \beta' \lambda_j \beta'}{(1 + \lambda_i \beta') (1 + \lambda_j \beta')} \right)^{1/2} \\
&= \left(1 - \frac{1}{1 + \lambda_i \beta'} \right)^{1/2} \left(1 - \frac{1}{1 + \lambda_j \beta'} \right)^{1/2},
\end{aligned} \tag{2.22}$$

où $COV[X_i, X_j]$ est la covariance entre X_i et X_j . Encore une fois, nous pouvons voir que $\lambda_i \beta'$ a le contrôle sur la corrélation : plus sa valeur est faible, plus la corrélation est nulle et plus elle est élevée, plus la corrélation est forte.

Nous pouvons en déduire que notre processus doit introduire une corrélation positive dans le but d'augmenter la variabilité dans le nombre d'arrivées par demi-heure. Ceci limite la flexibilité du modèle et implique qu'il sera difficile de connaître la nécessité de la corrélation pour représenter la réalité dans le centre d'appels. Toutefois, en comparant notre processus avec celui proposé par Jongbloed et Koole [31], nous avons un processus moins paramétrisé (26 paramètres vs 50 paramètres) qui nous permet d'ajouter une dépendance "nécessaire" (corrélations empiriques significatives) entre le nombre d'arrivées de demi-heures successives.

Cette famille de processus nous permet de considérer d'autres processus avec des paramètres supplémentaires. Nous pourrions avoir, par exemple, des α' , β' et/ou λ_k pour chaque jour de la semaine. Considérons une version fortement paramétrisée avec des paramètres spécifiques à chaque jour et une plus simple avec un ensemble de paramètres pour tous les jours. Parmi toutes les paramétrisations possibles entre ces deux versions, quelle est la meilleure ? Une façon de répondre à cette question est de tester l'utilité des paramètres. Il est possible d'estimer cette utilité en appliquant le test du ratio des fonctions de vrai-

semblance. Ce test décrit dans [16] nous permet de tester l’hypothèse que les paramètres θ optimaux se trouvent dans un sous-espace de paramètres possibles. Soit Θ , l’espace de paramètres le plus général (des paramètres spécifiques à chaque jour), et $\Theta_0 \in \Theta$ (les mêmes paramètres pour tous les jours). La statistique ω pour tester $H_0 : \theta \in \Theta_0$, est le ratio du maximum de la fonction de vraisemblance sous l’espace Θ_0 et du maximum sous l’espace Θ . Il est connu que $-2 \ln(\omega)$ suit asymptotiquement une loi chi-deux. Ce résultat va nous permettre d’évaluer si la valeur de ω observée est suffisamment petite pour rejeter l’hypothèse nulle [16]. Ainsi, nous aurons testé l’hypothèse que le processus avec le moins de paramètres peut aussi bien représenter les données qu’avec le plus de paramètres. Nous obtenons une statistique du test qui est loin de rejeter cette hypothèse, car $\omega = 0,9997$. Ceci suggère que le processus le moins paramétrisé s’ajuste aussi bien que celui avec une paramétrisation plus générale.

Finalement, nous pourrions tester et construire plusieurs autres processus pour représenter les arrivées d’appels entrants. Le développement de ces modèles plus complexes pourrait aider au niveau du réalisme. Nous pourrions, par exemple, modéliser les arrivées d’appels à l’aide d’une série chronologique. Le manque de temps et le format des données nous ont limité dans l’exploration de processus plus complexes. Peut-être que la suite du projet pourra étudier d’autres avenues.

2.1.2 La durée de service des appels entrants

La durée de service d’un appel entrant est définie comme la durée totale où un préposé est occupé avec le dossier d’un client. Ceci inclut le temps de mise en garde de l’appel, le temps de la conversation et le temps nécessaire à la mise à jour de dossiers. Nous allons dénoter la durée de service du client entrant c par $S_{e,c}$. Nous allons supposer que $S_{e,c}$ est une variable aléatoire suivant une loi de probabilité spécifique. L’ajustement de cette loi se fait généralement par l’observation de plusieurs durées de service individuelles. Malheureusement, nous n’avons pas pu obtenir ce type de données. Nous disposons seulement de la durée de service totale consacrée par les agents sur différentes demi-heures. Malgré cet obstacle, nous allons tenter d’ajuster des lois pour représenter $S_{e,c}$. Il est important de noter que les ajustements proposés se basent faussement sur le fait que les données donnent la durée de service totale des appels arrivant dans la demi-heure. Nous avons dû faire nos ajustements et nos tests avec cette petite erreur car nous ne disposons pas de meilleures

Demi-heure i	$\bar{S}_{e,i}$	Demi-heure i	$\bar{S}_{e,i}$
8h00-8h30	491,6	14h30-15h00	588,5
8h30-9h00	563,0	15h00-15h30	569,3
9h00-9h30	606,3	15h30-16h00	563,0
9h30-10h00	605,7	16h00-16h30	556,8
10h00-10h30	596,3	16h30-17h00	560,3
10h30-11h00	595,3	17h00-17h30	569,5
11h00-11h30	595,3	17h30-18h00	539,0
11h30-12h00	600,3	18h00-18h30	537,6
12h00-12h30	576,5	18h30-19h00	530,6
12h30-13h00	572,2	19h00-19h30	517,3
13h00-13h30	566,7	19h30-20h00	506,6
13h30-14h00	574,8	20h00-20h30	497,9
14h00-14h30	588,7	Toutes	572,1

Tableau 2.8 – Moyenne en secondes de la durée de service d’un appel entrant pour chaque demi-heure.

observations.

Dans un premier temps, nous allons faire l’hypothèse populaire que les durées individuelles $S_{e,c}$ pour chaque client c sont indépendantes et suivent une loi exponentielle de taux μ_e . Nous allons supposer que ce taux est constant durant toute la journée et de jour en jour. Il est assez facile de trouver un estimateur $\hat{\mu}_e$ pour le taux de la loi exponentielle en utilisant la méthode du maximum de vraisemblance : il suffit de calculer l’inverse de la moyenne empirique des durées de service individuelles [37], dénotée par \bar{S}_e . Soient X'_k le nombre d’appels entrants servis pour la k -ième demi-heure et Y_k la durée de service totale de ces appels. On définit :

$$\hat{\mu}_e = \frac{1}{\bar{S}_e} = \frac{\sum_{k=1}^d X'_k}{\sum_{k=1}^d Y_k},$$

où d est le nombre de demi-heures comprises dans les données. Nous avons $d = 5918$ et $\bar{S}_e = 572,0$ secondes.

Une deuxième modélisation possible est une durée de service suivant une loi exponentielle dont le taux varie au courant de la journée. Nous allons supposer que le taux est constant par demi-heure. En supposant que les demi-heures sont indépendantes entre elles, nous pouvons

remplacer la variable $S_{e,c}$ par des variables $S_{e,i,c}$ pour chaque client c de la demi-heure i . On suppose que ces $S_{e,i,c}$ sont indépendantes de loi exponentielle avec un taux $\mu_{e,i}$. Soient $X'_{i,j}$ le nombre d'appels entrants servis durant la i -ième demi-heure de la j -ième journée et $Y_{i,j}$ la durée totale de service de ces appels, nous pouvons utiliser la méthode de vraisemblance pour trouver un estimateur pour $\mu_{e,i}$:

$$\hat{\mu}_{e,i} = \frac{1}{\bar{S}_{e,i}} = \frac{\sum_{j=1}^r X'_{i,j}}{\sum_{j=1}^r Y_{i,j}},$$

où r est le nombre de journées observées dans les données et $\bar{S}_{e,i}$ est la moyenne empirique des durées de service individuelles pour la i -ième demi-heure. Le tableau 2.8 présente les résultats obtenus.

Le tableau 2.8 montre une moyenne des durées de service qui ne semble pas être constante durant la journée. On remarque qu'elle est plus faible lors des débuts et des fins de journée. Nous avons comparé les moyennes empiriques sur chaque jour de la semaine et il ne semble pas avoir de différence significative.

Nous allons maintenant tenter de tester l'hypothèse que $S_{e,i,c}$ suit une loi exponentielle de paramètres $\hat{\mu}_{e,i}$. L'agrégation des données nous forcent à utiliser un test indirect. Mais tout d'abord, nous allons définir les hypothèses nulles :

$$H_{0,i} \quad : \quad S_{e,i,1}, \dots, S_{e,i,\tilde{X}'_i} \text{ sont des variables aléatoires i.i.d.} \\ \text{de loi exponentielle avec paramètre } \hat{\mu}_{e,i},$$

où $\tilde{X}'_i = \sum_{j=1}^r X'_{i,j}$ est la somme de clients observée dans la demi-heure i . Pour tester indirectement $H_{0,i}$, nous procédons comme suit. On remarque que si $H_{0,i}$ est vraie alors cela implique que la loi de $Y_{i,j}$ conditionnelle à $X'_{i,j}$ est une loi gamma de paramètres $(X'_{i,j}, \hat{\mu}_{e,i})$ dont la fonction de répartition sera dénotée par $F_{i,j}$. $H_{0,i}$ implique que les variables aléatoires $U_{i,j} = F_{i,j}(Y_{i,j})$ suivent la loi uniforme $U(0, 1)$ et c'est ce que nous allons tester.

Soit \hat{F}_i la fonction de répartition empirique des $U_{i,j}$ de la demi-heure i , définie par

$$\hat{F}_i(u) = \frac{1}{r} \sum_{j=1}^r I[U_{i,j} \leq u] \quad \text{pour } i = 1, 2, \dots, 25, \quad (2.23)$$

où I est la fonction indicatrice et r est le nombre de jours. Nous allons utiliser le test de Kolmogorov-Smirnov (KS) [37] pour vérifier le réalisme de $H_{0,i}$. La statistique D_i de KS mesure la distance entre \hat{F}_i et la distribution de la loi uniforme :

$$D_i = \sup_{0 \leq u \leq 1} |\hat{F}_i - u| = \max(D_i^+, D_i^-), \quad (2.24)$$

où

$$D_i^+ = \max_{1 \leq j \leq r} \left\{ \frac{j}{r} - U_{i,j} \right\} \quad \text{et} \quad D_i^- = \max_{1 \leq j \leq r} \left\{ U_{i,j} - \frac{j-1}{r} \right\}. \quad (2.25)$$

Nous allons rejeter $H_{0,i}$ si la valeur prise par D_i est trop grande.

Le test de Kolmogorov-Smirnov ne peut pas s'appliquer directement, car nous ne connaissons pas la vraie valeur des μ_i . Ce test serait trop optimiste puisque les paramètres sont ajustés à l'aide des données. Ainsi, nous allons utiliser la méthode du "bootstrap" pour approximer la distribution de D_i sous $H_{0,i}$ [48]. Le tableau 2.9 donne les p -valeurs obtenues pour chaque demi-heure i avec 1000 répétitions. La valeur ϵ signifie que nous n'avons pas observé d'échantillons avec une distance supérieure à celle de D_i et qu'ainsi nous pouvons conclure que la vraie p -valeur doit être très petite. Bien que nous ayons obtenu quelques p -valeurs assez élevées, la grande majorité des valeurs observées impliquent le rejet des hypothèses $H_{0,i}$.

Demi-heure i	p_i	Demi-heure i	p_i
8h00-8h30	ϵ	14h30-15h00	0,001
8h30-9h00	0,69	15h00-15h30	0,15
9h00-9h30	0,04	15h30-16h00	0,17
9h30-10h00	0,13	16h00-16h30	0,68
10h00-10h30	0,001	16h30-17h00	0,01
10h30-11h00	0,05	17h00-17h30	0,02
11h00-11h30	ϵ	17h30-18h00	0,001
11h30-12h00	ϵ	18h00-18h30	ϵ
12h00-12h30	ϵ	18h30-19h00	ϵ
12h30-13h00	0,08	19h00-19h30	0,004
13h00-13h30	0,02	19h30-20h00	0,01
13h30-14h00	0,02	20h00-20h30	ϵ
14h00-14h30	ϵ	Toutes	ϵ

Tableau 2.9 – Approximation de la p -valeur du test statistique indirect pour la représentation de la durée de service d'un appel entrant comme une variable aléatoire de loi exponentielle variant par demi-heure.

Il est intéressant de noter que nous avons simplement testé que la somme des durées de service pour chaque demi-heure donnée, conditionnelle au nombre d'appels servis, suit une

loi gamma. Cette hypothèse est plus faible que celle qui suppose que l'on a des durées de service individuelles i.i.d exponentielles. En fait, nous avons réussi à rejeter une hypothèse qui devrait être plus difficile à rejeter.

Dans le rapport [7], on observe que les durées de service des appels sortants semblent suivre une loi lognormale. Nous allons maintenant supposer que cette loi caractérise bien la durée de service des appels entrants. Pour estimer ces paramètres, nous allons utiliser la méthode des moments. Cette méthode ajuste une loi qui présente les mêmes moments que l'on observe dans les données. Le k -ième moment de S_e est défini par :

$$m_k = E(S_e^k). \quad (2.26)$$

Si $S_{e,1}, S_{e,2}, \dots, S_{e,z}$ sont des variables i.i.d provenant de cette loi, on définit le k -ième moment empirique par :

$$\hat{m}_k = \frac{1}{z} \sum_{i=1}^z S_{e,i}^k. \quad (2.27)$$

La méthode des moments estime les paramètres en les représentant à l'aide des moments d'ordre le plus faible possible et en substituant les moments empiriques dans ces expressions. Les paramètres μ et σ^2 de la loi lognormale, définie par la fonction de densité

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp \frac{-(\ln x - \mu)^2}{2\sigma^2}, \text{ où } x > 0,$$

peuvent s'exprimer par les deux premiers moments de la loi :

$$\mu = 2 \ln m_1 - \frac{1}{2} \ln m_2 \quad \text{et} \quad \sigma^2 = \ln m_2 - 2 \ln m_1. \quad (2.28)$$

Pour appliquer cette méthode, il faut alors calculer les deux premiers moments empiriques. L'agrégation des données complexifie quelque peu le calcul de ces moments. Le premier moment empirique est facile à trouver :

$$\hat{m}_1 = \frac{\sum_{k=1}^d X'_k \bar{Y}_k}{\sum_{k=1}^d X'_k}, \quad (2.29)$$

où \bar{Y}_k est la moyenne des durées de service pour la k -ième demi-heure. Pour ce qui est du calcul de \hat{m}_2 , il se fera par l'entremise du calcul de la variance de S_e . Nous savons que

$$\text{VAR}(S_e) = E(S_e^2) - E(S_e)^2. \quad (2.30)$$

Donc si nous pouvons estimer $\text{VAR}(S_e)$ et $E(S_e)^2$, nous allons pouvoir obtenir un estimateur pour $E(S_e^2)$, i.e., pour le deuxième moment de la loi. Nous allons trouver un estimateur en

supposant que X'_k est indépendant de \bar{Y}_k . Ceci n'est pas trop dangereux car nous avons beaucoup d'observations dans le calcul de \bar{Y}_k . Ces hypothèses simplifient le calcul d'un estimateur.

Proposition 2.1.1 *L'estimateur*

$$\hat{V}^2 = \frac{1}{d-1} \sum_{k=1}^d X'_k (\bar{Y}_k - \hat{m}_1)^2 \quad (2.31)$$

est non-biaisé pour $VAR(S_e)$.

Preuve (avec $\tilde{X}' = \sum_{k=1}^d X'_k$) :

$$\begin{aligned} \mathbb{E} [\hat{V}^2] &= \frac{1}{d-1} \mathbb{E} \left[X'_1 (\bar{Y}_1 - \hat{m}_1)^2 + X'_2 (\bar{Y}_2 - \hat{m}_1)^2 + \dots + X'_d (\bar{Y}_d - \hat{m}_1)^2 \right] \\ &= \frac{1}{d-1} \mathbb{E} \left[X'_1 \bar{Y}_1^2 + X'_2 \bar{Y}_2^2 + \dots + X'_d \bar{Y}_d^2 \right. \\ &\quad \left. - 2X'_1 \bar{Y}_1 \hat{m}_1 - 2X'_2 \bar{Y}_2 \hat{m}_1 - \dots - 2X'_d \bar{Y}_d \hat{m}_1 + \tilde{X}' \hat{m}_1^2 \right] \\ &= \frac{1}{d-1} \mathbb{E} \left[X'_1 \bar{Y}_1^2 + X'_2 \bar{Y}_2^2 + \dots + X'_d \bar{Y}_d^2 \right. \\ &\quad \left. - 2\hat{m}_1 (X'_1 \bar{Y}_1 + X'_2 \bar{Y}_2 + \dots + X'_d \bar{Y}_d) + \tilde{X}' \hat{m}_1^2 \right] \\ &= \frac{1}{d-1} \mathbb{E} \left[X'_1 \bar{Y}_1^2 + X'_2 \bar{Y}_2^2 + \dots + X'_d \bar{Y}_d^2 - \tilde{X}' \hat{m}_1^2 \right] \\ &= \frac{1}{d-1} \left(\mathbb{E} [X'_1 \bar{Y}_1^2] + \mathbb{E} [X'_2 \bar{Y}_2^2] + \dots + \mathbb{E} [X'_d \bar{Y}_d^2] - \mathbb{E} [\tilde{X}' \hat{m}_1^2] \right) \\ &= \frac{1}{d-1} \left(\mathbb{E} [X'_1] \mathbb{E} [\bar{Y}_1^2] + \mathbb{E} [X'_2] \mathbb{E} [\bar{Y}_2^2] + \dots + \mathbb{E} [X'_d] \mathbb{E} [\bar{Y}_d^2] - \mathbb{E} [\tilde{X}'] \mathbb{E} [\hat{m}_1^2] \right) \\ &= \frac{1}{d-1} \left(\mathbb{E} [X'_1] \left(m_1^2 + \frac{\sigma^2}{\mathbb{E} [X'_1]} \right) + \mathbb{E} [X'_2] \left(m_1^2 + \frac{\sigma^2}{\mathbb{E} [X'_2]} \right) + \dots \right. \\ &\quad \left. + \mathbb{E} [X'_d] \left(m_1^2 + \frac{\sigma^2}{\mathbb{E} [X'_d]} \right) - \mathbb{E} [\tilde{X}'] \left(m_1^2 + \frac{\sigma^2}{\mathbb{E} [\tilde{X}']} \right) \right) \\ &= \frac{1}{d-1} \left(\mathbb{E} [\tilde{X}'] m_1^2 + d\sigma^2 - \mathbb{E} [\tilde{X}'] m_1^2 - \sigma^2 \right) \\ &= \sigma^2. \quad \square \end{aligned}$$

Avec les estimateurs \hat{m}_1 et \hat{V}^2 , nous pouvons estimer par la méthode des moments les paramètres de la loi lognormale. Nous obtenons respectivement les valeurs 5,874 et 0,948 pour $\hat{\mu}$ et $\hat{\sigma}^2$. Il nous resterait à appliquer un test statistique pour vérifier le réalisme de cette modélisation. Pour l'instant, nous n'avons pas réussi à développer un test indirect

comme nous l'avons proposé dans le cas de la loi exponentielle. Ceci est dû à la complexité d'un tel test avec une loi lognormale.

La sous-section 2.2.1 montre aussi que la distribution des durées de service des appels sortants est semblable à celle d'une loi gamma. Nous allons tester cette possibilité pour les appels entrants. Nous pouvons utiliser encore une fois la méthode des moments pour ajuster les paramètres de cette loi. Les paramètres de la fonction de densité d'une loi gamma que l'on retrouve dans l'équation (2.6) sont estimés par :

$$\hat{\beta} = \frac{\hat{V}^2}{\hat{m}_1} \quad \text{et} \quad \hat{\alpha} = \frac{\hat{m}_1^2}{\hat{V}^2}. \quad (2.32)$$

Lors de l'ajustement d'une lognormale, nous avons décrit comment calculer les valeurs de \hat{m}_1 et \hat{V}^2 . Ainsi, nous pouvons obtenir facilement les valeurs des paramètres $\hat{\beta}$ et $\hat{\alpha}$ d'abord en supposant qu'ils sont constants pour toute la journée puis en supposant qu'ils varient entre chaque demi-heure. Le tableau 2.10 présente les résultats obtenus. Contrairement à la loi lognormale, nous avons pu développer un test indirect. Nous savons que la somme de variables i.i.d. $S_{e,1}, S_{e,2}, \dots, S_{e,z}$ avec $S_{e,k} \sim \text{gamma}(\alpha, \beta)$ suit une loi $\text{gamma}(z\alpha, \beta)$. Cette affirmation nous permet de dire que la somme des durées de service pour la demi-heure i du jour j , conditionnelle au nombre d'appels servis dans cette demi-heure, suit une loi gamma de paramètres $(X'_{i,j}\alpha, \beta)$. Nous avons testé cette hypothèse en appliquant le test décrit pour la loi exponentielle avec les paramètres de la loi gamma. Les p -valeurs associées à chaque demi-heure ont été obtenues par une méthode de "bootstrap" et sont présentées dans le tableau 2.10. On peut remarquer que la p -valeur d'une modélisation unique pour toutes les demi-heures est assez faible pour rejeter cette hypothèse. Cependant, les p -valeurs d'une modélisation par demi-heure ne permettent pas de rejeter indirectement cette autre hypothèse. Ainsi, nous pouvons supposer que les durées de service des appels entrants peuvent suivre une loi gamma de paramètres variant par demi-heure. Vu que nous n'avons pas pu vérifier le réalisme de la loi lognormale, nous allons utiliser la loi gamma dans notre modèle le plus réaliste.

2.1.3 Les abandons

Dans la file d'attente d'un centre d'appels, il peut se produire des abandons, c'est-à-dire que le client n'a plus la patience d'attendre et il décide de terminer l'appel sans avoir reçu son service. Nous allons dénoter par $A_{e,c}$ la durée entre l'instant où le client c d'un

Demi-heure i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i	Demi-heure i	$\hat{\alpha}_i$	$\hat{\beta}_i$	p_i
8h00-8h30	1,374	357,4	0,55	14h30-15h00	0,702	838,4	0,34
8h30-9h00	0,924	608,9	0,78	15h00-15h30	0,782	727,9	0,89
9h00-9h30	0,956	634,1	0,50	15h30-16h00	0,776	725,2	0,73
9h30-10h00	0,807	750,5	0,65	16h00-16h30	0,821	678,3	0,93
10h00-10h30	0,735	811,3	0,21	16h30-17h00	0,565	992,3	0,76
10h30-11h00	0,706	843,6	0,61	17h00-17h30	0,583	970,4	0,73
11h00-11h30	0,700	888,9	0,60	17h30-18h00	0,583	917,0	0,66
11h30-12h00	0,533	1126,2	0,58	18h00-18h30	0,496	1080,2	0,16
12h00-12h30	0,664	863,9	0,37	18h30-19h00	0,487	1089,0	0,44
12h30-13h00	0,740	770,0	0,89	19h00-19h30	0,506	1021,4	0,35
13h00-13h30	0,465	1218,0	0,38	19h30-20h00	0,536	944,1	0,79
13h30-14h00	0,615	935,0	0,93	20h00-20h30	0,505	986,7	0,34
14h00-14h30	0,697	844,9	0,56	Toutes	0,633	903,2	0,09

Tableau 2.10 – Paramètres estimés ($\hat{\alpha}_i$ et $\hat{\beta}_i$) et p -valeurs d'un test statistique pour la représentation de la durée de service d'un appel entrant comme une variable aléatoire de loi gamma variant par demi-heure.

appel entrant débute son attente et l'instant où il n'a plus la patience d'attendre. Cette durée sera souvent référée comme la durée d'attente acceptée. La modélisation des $A_{e,c}$ peut être importante. C'est particulièrement le cas lorsqu'elle influence la qualité de service à satisfaire.

Il est malheureux que Bell n'ait pas réussi à nous fournir les données pertinentes pour modéliser cet aspect. Les seules informations disponibles sont le nombre d'abandons pour chaque demi-heure et la répartition des durées d'attente pour les clients ayant abandonné. Cette répartition ne suffit pas puisqu'elle est conditionnelle à ce que le client abandonne. Elle n'est pas vraiment représentative vu le peu d'abandons et ainsi, elle ne nous permet pas de déduire une répartition du temps d'attente accepté pour tous les clients (abandons ou pas).

Dans nos différents modèles, nous allons toutefois faire certaines hypothèses sur ce temps d'attente accepté. Tout d'abord, nous allons supposer que A_e joue un rôle seulement lorsqu'une attente est nécessaire. Dans certains modèles, nous allons faire la supposition que

les $A_{e,c}$ sont i.i.d de loi exponentielle avec un taux variable et/ou qu'elles ont une forte probabilité d'avoir une durée d'attente acceptée nulle, i.e., la fonction de répartition a un saut pour la valeur 0.

Le tableau 2.11 donne un intervalle de confiance de la proportion, en pourcentage, d'abandons sur le nombre d'appels entrants reçus durant 216 jours de semaine au centre d'appels étudié. Nous remarquons que le pourcentage d'abandons varie significativement tout au long de la journée. On observe des valeurs plus élevées pour les débuts et les fins de journée. Ensuite, nous avons tenté d'observer l'influence du jour de la semaine sur le pourcentage d'abandons et nous n'avons pas remarqué de différence à l'exception du samedi. Cette journée présente un pourcentage nettement plus élevé : nous avons un pourcentage d'abandons de 2,7 % du lundi au vendredi tandis que nous avons 3,9 % durant la journée du samedi. Nous pouvons peut-être expliquer ce phénomène par un manque de patience pour le client appelant le samedi ou par un temps d'attente moyen supérieur aux autres jours de la semaine. Il nous est impossible de confirmer ces explications puisque les données disponibles ne nous permettent pas d'estimer le temps d'attente moyen généré par le centre.

Demi-heure	% abandons	Demi-heure	% abandons
8h00-8h30	4,9 ± 1,7	14h30-15h00	1,5 ± 0,3
8h30-9h00	2,0 ± 0,5	15h00-15h30	1,3 ± 0,2
9h00-9h30	3,3 ± 1,3	15h30-16h00	1,2 ± 0,2
9h30-10h00	2,9 ± 0,8	16h00-16h30	1,5 ± 0,3
10h00-10h30	3,1 ± 0,6	16h30-17h00	2,1 ± 0,4
10h30-11h00	2,9 ± 0,7	17h00-17h30	2,0 ± 0,3
11h00-11h30	3,6 ± 0,8	17h30-18h00	2,7 ± 0,4
11h30-12h00	3,6 ± 0,8	18h00-18h30	3,1 ± 0,8
12h00-12h30	3,5 ± 0,7	18h30-19h00	3,6 ± 0,7
12h30-13h00	4,7 ± 1,1	19h00-19h30	3,8 ± 1,2
13h00-13h30	1,3 ± 0,4	19h30-20h00	3,1 ± 1,0
13h30-14h00	1,9 ± 0,5	20h00-20h30	5,5 ± 1,1
14h00-14h30	1,4 ± 0,3	Toutes	2,7 ± 0,1

Tableau 2.11 – Intervalle de confiance du pourcentage des appels entrants qui abandonnent avant de se faire servir.

2.1.4 La qualité de service

La principale qualité de service que l'on veut satisfaire est la règle 80-20, c'est-à-dire qu'il doit avoir plus de 80 % des clients entrants qui attendent moins de 20 secondes. Il est important de noter que cette qualité de service n'est pas un élément que l'on doit modéliser directement. Cependant, on doit en faire l'analyse avec les données recueillies sur le centre d'appels afin de connaître la qualité de service sous la politique de gestion actuelle. Ceci sera utile pour évaluer le réalisme des modèles proposés en comparant leurs mesures de qualité de service.

Son calcul à partir des données est assez simple. Nous allons dénoter par Q_k la qualité de service observée durant la k -ième demi-heure observée, c'est-à-dire le pourcentage des appels entrants reçus au cours de cette demi-heure dont l'attente n'a pas dépassé 20 secondes :

$$Q_k = \frac{((X'_k + B_k) - (X20'_k + B20_k)) \times 100}{X'_k + B_k}, \quad (2.33)$$

où pour la demi-heure k , B_k est le nombre d'abandons, $X20'_k$ est le nombre d'appels entrants servis après 20 secondes d'attente et $B20_k$ est le nombre d'abandons après plus de 20 secondes d'attente. Ces informations nous permettent de calculer la qualité de service moyenne \bar{Q} sur l'ensemble des données :

$$\bar{Q} = \frac{\sum_{k=1}^d (X'_k + B_k) Q_k}{\sum_{k=1}^d (X'_k + B_k)}.$$

Nous pouvons faire le calcul de la qualité de service observée dans chacune des demi-heures. Dans ce cas, on retient seulement la qualité de service de la demi-heure qui nous intéresse. Le tableau 2.12 présente la qualité de service moyenne \bar{Q}_i pour chaque demi-heure i des jours réguliers de la semaine, i.e., sans les samedi et les 5 jours spéciaux déjà identifiés.

Le tableau 2.12 montre des valeurs assez élevées pour la qualité de service. On pourrait même dire que ces valeurs sont un peu trop grandes dans le cas où nous devons uniquement respecter la règle 80-20. Ces résultats sont peut-être dûs à l'utilisation d'une politique de gestion trop conservatrice. Par exemple, il se peut que le centre manque d'agressivité au niveau de la composition d'appels sortants ou sur-planifie le nombre nécessaire d'agents. L'amélioration des politiques de gestion est justement un des buts du projet. Il semble que nous pouvons améliorer la performance du centre d'appels en étant plus agressif tout en respectant la qualité de service définie par la règle 80-20. Nous disposons d'une bonne marge

Demi-heure i	\bar{Q}_i	Demi-heure i	\bar{Q}_i
8h00-8h30	81,4 %	14h30-15h00	93,5 %
8h30-9h00	86,5 %	15h00-15h30	94,5 %
9h00-9h30	82,6 %	15h30-16h00	94,2 %
9h30-10h00	81,2 %	16h00-16h30	91,9 %
10h00-10h30	78,1 %	16h30-17h00	89,0 %
10h30-11h00	82,1 %	17h00-17h30	92,8 %
11h00-11h30	77,7 %	17h30-18h00	86,4 %
11h30-12h00	80,1 %	18h00-18h30	84,2 %
12h00-12h30	78,0 %	18h30-19h00	84,8 %
12h30-13h00	75,1 %	19h00-19h30	85,1 %
13h00-13h30	92,4 %	19h30-20h00	89,8 %
13h30-14h00	91,1 %	20h00-20h30	89,4 %
14h00-14h30	92,9 %	Toutes	86,1 %

Tableau 2.12 – Pourcentage des appels entrants (par demi-heure) ayant attendu moins de 20 secondes avant de se faire servir ou abandonner pour les jours réguliers de la semaine.

de manoeuvre compte tenu que la qualité de service moyenne se situe pour le moment à 86,1 %.

Quelques rencontres avec des personnes ressources de chez Bell nous ont permis de trouver un comportement qui pourrait expliquer en partie la trop bonne qualité de service observée. Nous croyons que les gestionnaires d'accès sur-réagissent lorsqu'ils observent une mauvaise qualité de service à court terme. Nous allons démontrer dans une prochaine section le danger de cette sur-réaction.

Nous avons fait l'analyse de la qualité de service observée pour chaque jour de la semaine. Nous obtenons sensiblement les mêmes résultats pour les jours entre le lundi et le vendredi. Cependant, nous notons un comportement différent pour la journée du samedi. Nous obtenons une qualité de service plus basse mais toujours acceptable : 80,8 % pour le samedi vs 86,1 % pour l'ensemble des autres journées.

2.2 Les appels sortants

Dans cette section, nous allons analyser des aspects associés aux appels sortants. Nous allons principalement nous attarder à la modélisation des durées de service pour les appels sortants et sur le processus de composition de ces appels. Nous avons deux bases de données sur les appels sortants. La première base couvre 190 jours entre le 6 juin 2001 et le 2 mars 2002 et elle fournit des observations agrégées par demi-heure. La deuxième base présente les durées de service individuelles, mais elle ne couvre que 121 jours entre le 26 septembre 2001 et le 2 mars 2002.

2.2.1 La durée de service des appels sortants

Nous définissons la durée de service de ces appels de la même manière que celle des appels entrants, i.e., on considère la durée totale où l'agent est occupé par le client. Nous allons dénoter par $S_{s,c}$ la durée de service requise pour le client d'appel sortant c . Nous allons supposer que les $S_{s,c}$ sont des variables aléatoires i.i.d. qui suivent une loi de probabilité précise. Avant de proposer une loi, il serait bon d'observer la distribution empirique et la fonction de densité empirique des $C = 54444$ durées individuelles. Les figures 2.3 et 2.4 présentent ces fonctions.

La fonction de répartition empirique a quelques similitudes avec celle d'une loi exponentielle bien que son ascension se fait plus doucement. Il est facile de voir que cette hypothèse est peu réaliste lorsque l'on observe la fonction de densité empirique. Cette fonction ressemble plus à la densité d'une lognormale ou encore à celle d'une gamma. C'est justement avec ces lois que nous allons tenter de modéliser les durées de service pour les appels sortants. Pour chacune des lois, nous allons estimer les paramètres qui s'ajustent le mieux aux données. Ensuite, nous allons tenter d'éprouver chacune des hypothèses à l'aide de tests statistiques.

Bien que nous ayons vu que la fonction de densité est loin d'être semblable à celle d'une loi exponentielle, nous allons estimer un taux, à l'aide de la méthode du maximum de vraisemblance, qui s'ajuste bien aux données. L'estimateur $\hat{\mu}_s$ pour le taux μ_s de la loi se calcule en inversant la moyenne \bar{S}_s des durées de service observées.

Tout dépendant de la base de données, nous trouvons $\bar{S}_s = 440,2$ secondes avec la

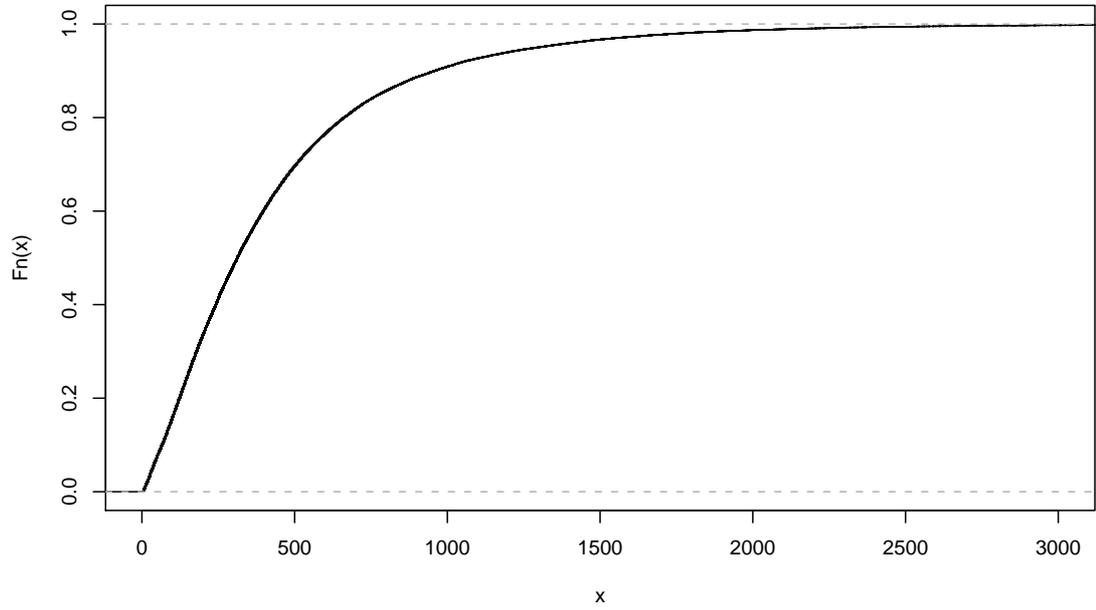


Figure 2.3 – Fonction de répartition empirique pour les durées de service d'appels sortants.

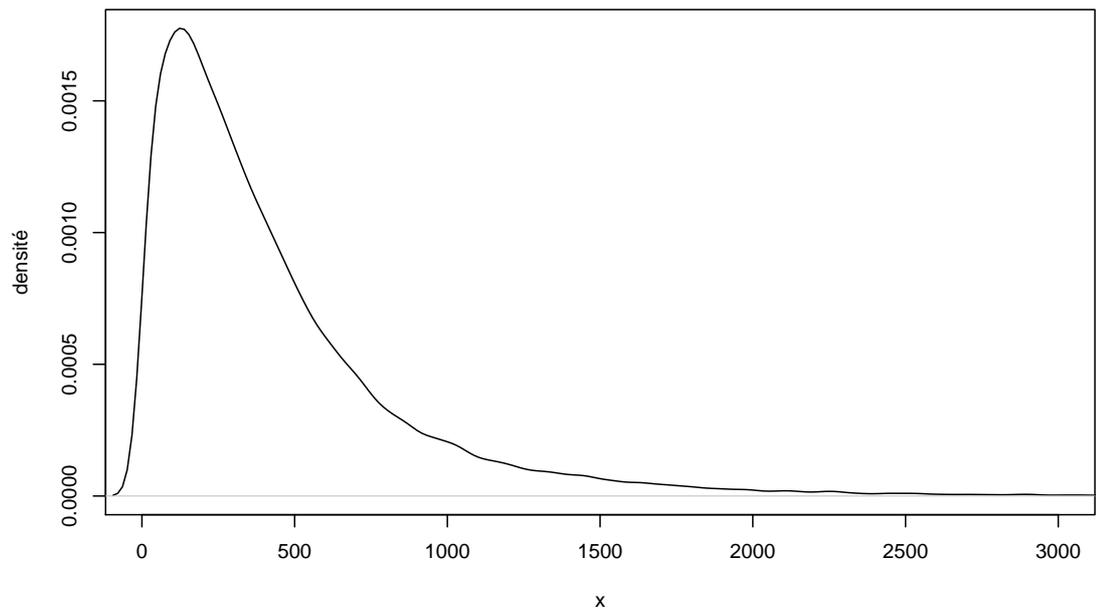


Figure 2.4 – Fonction de densité empirique pour les durées de service d'appels sortants.

première et $\bar{S}_S = 446,2$ secondes avec la deuxième. La deuxième base a l'avantage de contenir plus d'observations, mais la première nous permet d'estimer facilement une variance empirique \hat{S}^2 pour S_S :

$$\hat{S}^2 = \frac{1}{C-1} \sum_{c=1}^C (S_{S,c} - \bar{S}_S)^2, \quad (2.34)$$

où C est le nombre de durées observées. Sous l'hypothèse de variables aléatoires i.i.d exponentielles, nous savons que la racine carrée de la variance des durées devrait être égale à la moyenne. L'estimation obtenue pour $\sqrt{\hat{S}^2}$ est égale à 452,3, ce qui n'est pas loin de la moyenne estimée. Malgré cette observation, nous avons remarqué que la fonction de densité empirique était assez différente de celle d'une loi exponentielle (figure 2.4). Pour s'assurer que nous n'avons pas cette loi de probabilité, nous avons fait le test de Kolmogorov-Smirnov dont le calcul de la p -valeur se fait à l'aide de la méthode "bootstrap". La p -valeur obtenue est inférieure à 0,0001 et ainsi, nous devons rejeter cette hypothèse.

La deuxième loi étudiée est la gamma, car il semble que la fonction de densité de cette loi pourrait bien s'ajuster à la densité empirique de la figure 2.4. Nous avons déjà introduit la méthode des moments qui estime les paramètres α et β avec les deux premiers moments empiriques (équation (2.32)). Avec les données, nous obtenons par cette méthode les valeurs $\hat{\alpha} = 0,947$ et $\hat{\beta} = 464,8$. L'estimateur de α montre que nous sommes assez près d'une loi exponentielle et que le mode est zéro. Un mode de zéro est contradictoire à la fonction de densité empirique observée. Nous avons toutefois testé cette hypothèse avec le test de Kolmogorov-Smirnov. La p -valeur, obtenue par "bootstrap", est inférieure à 0,001. Encore une fois, nous devons rejeter cette modélisation.

La prochaine loi étudiée est souvent utilisée pour modéliser une durée associée à une tâche. La loi Weibull est définie par les paramètres α et β , ainsi que par la fonction de densité :

$$f(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^\alpha} \quad \text{si } x > 0. \quad (2.35)$$

Les estimateurs de vraisemblance pour les paramètres α et β se calculent facilement par une méthode numérique [37]. Nous avons obtenu des valeurs de 1,0696 pour l'estimation de α et de 452,52 pour l'estimation de β . Nous avons effectué le test de Kolmogorov-Smirnov avec ces paramètres. Cependant, nous devons rejeter l'hypothèse car la p -valeur associée est inférieure à 0,001.

La loi de probabilité lognormale est une autre loi qui nous inspirait confiance afin d'ob-

tenir une bonne modélisation. Nous allons estimer les paramètres de la loi lognormale en utilisant les estimateurs de vraisemblance maximaux (EVM). Dans ce cas particulier, il est plus logique d'utiliser la première base de données (durée de service individuelle) car elle nous permet d'obtenir des estimateurs avec moins de variance. Cette loi de probabilité définie par les paramètres μ et σ^2 . La méthode pour calculer $\hat{\mu}_1$, l'estimateur EVM de μ et $\hat{\sigma}_1^2$, l'estimateur EVM de σ^2 , est décrite ci-dessous [37] :

$$\hat{\mu}_1 = \frac{\sum_{c=1}^C \ln S_{s,c}}{C}, \quad (2.36)$$

$$\hat{\sigma}_1^2 = \frac{\sum_{c=1}^C (\ln S_{s,c} - \hat{\mu}_1)^2}{C - 1}. \quad (2.37)$$

Cette méthode revient à ajuster une loi normale aux valeurs $\ln S_{s,c}$. Nous obtenons $\hat{\mu}_1 = 5,60$ et $\hat{\sigma}_1^2 = 1,22$. Un des problèmes avec cette méthode est qu'elle n'est pas sans biais pour la moyenne des durées de service. Il est facile d'observer ce biais avec les estimateurs trouvés. La moyenne des durées de service que l'on obtient avec une loi lognormale de paramètres $\hat{\mu}_1$ et $\hat{\sigma}_1^2$ est égale à $\exp\{\hat{\mu}_1 + \hat{\sigma}_1^2/2\} = 497,7$ secondes, ce qui est loin de la moyenne empirique (440,2 secondes).

L'EVM n'est pas la seule méthode d'estimation possible. La méthode des moments est une alternative qui nous permet de respecter la moyenne empirique. Nous avons déjà décrit comment utiliser cette méthode pour estimer les paramètres de la loi lognormale. Cette deuxième méthode nous donne les estimateurs $\hat{\mu}_2 = 5,73$ et $\hat{\sigma}_2^2 = 0,72$. Comme prévu, cette méthode nous permet d'avoir une loi lognormale dont les paramètres respectent la moyenne empirique. Nous observons le même phénomène pour la variance des durées de service. La variance obtenue avec $\hat{\mu}_1$ et $\hat{\sigma}_1^2$ est deux fois trop grande par rapport à la variance empirique tandis que la variance avec $\hat{\mu}_2$ et $\hat{\sigma}_2^2$ respecte évidemment la variance empirique car elle est fonction des deux premiers moments.

La figure 2.5 présente la fonction de densité empirique et les fonctions de densité pour la loi lognormale avec les deux méthodes d'estimation. Même si nous obtenons une forme semblable pour toutes les fonctions, il est facile de voir qu'elles ne s'ajustent pas bien aux données. Nous avons appliqué le test de Kolmogorov-Smirnov pour des paramètres inconnus ("bootstrap"). Les p -valeurs obtenues sont très petites ($< 0,001$) et elles nous obligent à rejeter cette hypothèse qu'importe la façon d'estimer les paramètres.

L'ajustement d'une loi de probabilité sur $C = 54444$ observations est loin d'être une tâche facile. Avec une valeur aussi élevée, il semble que n'importe quelle loi standard sera

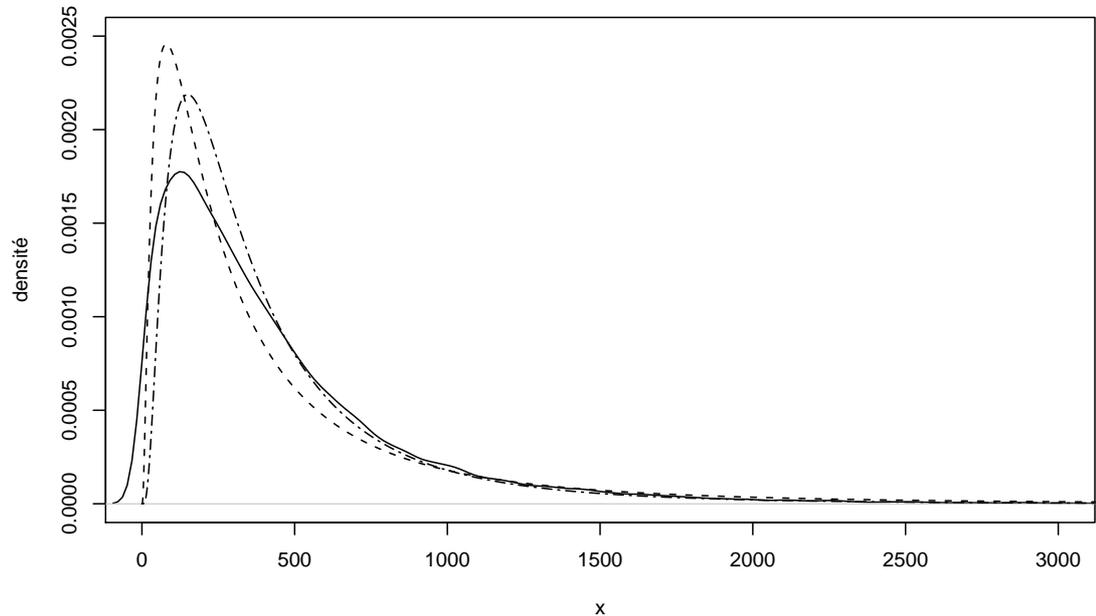


Figure 2.5 – Fonction de densité empirique pour les durées de service d’appels sortants (ligne pleine) vs celles obtenues en calculant les EVM (ligne pointillée - - -) et les EMM (ligne pointillée -.-) d’une loi lognormale.

rejetée : le test devient trop puissant. Il serait peut-être plus facile d’ajuster une loi sur différents sous-groupes de données. Le groupement par intervalles de temps pourrait être une bonne solution d’autant plus qu’il nous a donné de bons résultats pour la modélisation de certains aspects du centre d’appels. Nous allons supposer que les durées de service d’appels sortants suivent une loi lognormale dont les paramètres varient par demi-heure. Il suffit de calculer les estimateurs pour chaque demi-heure. Les p -valeurs obtenues nous obligent encore une fois à rejeter la modélisation.

Nous ne nous attendions pas à avoir de si mauvais résultats avec la loi lognormale. Dans un rapport d’étape pour Bell [7], nous avons trouvé que cette loi s’ajustait bien aux 6066 observations disponibles à cette époque, avec une p -valeur de 0,7924. Cette p -valeur est élevée et on croyait bien que la modélisation par la loi lognormale était bonne. L’explication de ce rejet pourrait être que le centre d’appels étudié est en constante évolution. Il n’y a même pas un an qu’il a ouvert ses portes et plusieurs ajustements ont dû être faits. Les

premières données que nous avons analysées venaient d'un groupe d'agents et de clients inexpérimentés. La moyenne des durées de service dans [7] était de 381 secondes tandis que nous avons maintenant une moyenne de 440 secondes. De plus, on trouve que la proportion d'appels ayant mené à une vente a considérablement augmentée dans les nouvelles données. Elle est passée de 1,75 % des appels à 3,1 %. Cette augmentation est assez significative lorsqu'on sait que ces appels demandent beaucoup plus de temps de service : moyenne de 1283 secondes vs 413 secondes pour les autres appels. Cette instabilité dans le centre nous oblige à modifier notre modélisation du rapport [7].

Il existe aussi les distributions du système de Johnson qui sont définies par quatre paramètres [37]. Ces distributions pourraient peut-être bien s'appliquer à notre problème. Le manque de temps nous a empêché d'explorer cette alternative. Nous en faisons tout de même mention car elle pourrait faire partie de la suite du projet. Il existe plusieurs références pertinentes sur l'estimation de ces paramètres [28, 54, 50]. Une autre alternative serait d'étudier les lois de type λ -généralisé [36]. Encore une fois, cette possibilité n'a pas été expérimentée, mais elle pourrait faire partie de la suite du projet.

Dans notre analyse sur cette durée de service, nous avons observé une grande différence entre la moyenne des appels menant à une vente (1283 secondes) et la moyenne des autres types d'appels (413 secondes). Nous avons pensé d'exploiter cette observation en divisant la modélisation en ces deux types d'appels. Dans un premier temps, nous avons essayé d'ajuster une loi pour les durées d'appels menant à une vente. Quelques expériences nous ont fait découvrir que nous ne pouvons pas rejeter l'hypothèse que ces durées suivent une loi gamma de paramètres $\beta = 538,1$ et $\alpha = 2,385$. La p -valeur que nous avons trouvée est égale à 0,144. Malgré ce premier bon ajustement, nous n'avons pas pu trouver une autre loi s'ajustant bien avec la durée des appels ne menant pas à une vente. Ce type de catégorisation semble une bonne idée qui pourrait être étudiée dans la suite du projet.

Jusqu'à présent, nous sommes loin d'avoir obtenu une bonne modélisation pour cette durée de service. L'ajustement "parfait" semble difficile à obtenir. Ce problème pourrait être réglé en utilisant une fonction de répartition empirique construite par noyaux. La librairie UNURAN (Universal Non-Uniform RANdom number generators) permet d'utiliser une fonction qui génère des variables suivant une telle fonction de répartition [42]. L'idée de cette méthode consiste à choisir une valeur parmi les données empiriques et de lui additionner un certain bruit. Ceci est équivalent à générer à partir d'une densité à noyaux dont

la forme générale est définie par :

$$f_n(x) = \frac{1}{nh} \sum_{i=1}^n k((x - x_i)/h), \quad (2.38)$$

où k définit une fonction de densité et h est un facteur de lissage. Bref, son utilisation revient à décider d'une fonction de densité pour le bruit. La librairie UNURAN fixe initialement un bruit Gaussien. En se référant à la figure 2.4, nous voyons bien que l'utilisation d'un tel bruit n'est pas l'idéal pour la représentation de la queue de la fonction de densité. Nous avons pensé à une alternative qui serait d'ajuster une densité à noyaux avec un bruit Gaussien sur le logarithme naturel des données. Ce choix est logique car nous avons vu que la densité empirique était assez près de celle d'une loi lognormale et ainsi, le logarithme des données doit être assez près d'une loi normale. Ceci va nous permettre d'avoir une meilleure représentation de la queue de la fonction de densité empirique. De plus, sa génération demeure simple car il suffit de calculer l'exponentielle d'une valeur provenant de la densité à noyaux avec un bruit Gaussien. Nous allons exploiter cette possibilité dans nos modèles.

2.2.2 Le processus de composition d'appels sortants

Le processus de composition est un élément important à modéliser. Il est le principal responsable de la performance d'un centre d'appels en mode mixte, i.e., c'est lui qui gère indirectement le trafic d'appels sortants à servir. Il exerce une influence sur le taux d'occupation des agents et sur la qualité de service du centre. Une bonne compréhension de cet aspect nous permettrait de modéliser correctement le processus de composition actuel. Nous pourrions, par la suite, proposer et comparer de nouvelles politiques de composition. Cette tâche de modélisation n'est pas si facile qu'elle peut le sembler. Comme nous l'avons expliqué dans l'introduction, la composition se fait par l'entremise d'un système développé par la compagnie Avaya. Bell connaît peu d'informations sur le fonctionnement interne et l'algorithme de décisions de ce système. Malheureusement, Bell est notre seule source d'informations pour modéliser ce processus. Ceci implique que notre tâche se ramène à modéliser un système qui nous apparaît comme une boîte noire.

Cet obstacle ne nous a pas vraiment empêché de développer des processus de composition qui représentent assez bien ce que l'on retrouve dans le vrai centre d'appels, c'est-à-dire qui montrent des mesures de performance semblables à celles observées dans la réalité. La simulation fut d'une grande utilité pour développer et surtout comparer les différents

processus de composition.

Maintenant, nous allons présenter un résumé de statistiques reliées à ce système de composition automatique. Nous étudions le nombre d’appels composés, le nombre d’appels rejoints et le nombre de “mismatches”. Un “mismatch” est défini par un client rejoint ne pouvant pas être servi immédiatement. Pour chacun de ces aspects, nous allons présenter une analyse sous la forme d’un intervalle de confiance, en supposant que leur moyenne tend vers la loi normale (théorème centrale limite [45]). Les résultats se retrouvent dans les tableaux 2.13 et 2.14.

Demi-heure i	appels composés	clients rejoints	service requis	“mismatches”	κ_i
14h00-14h30	86,8 \pm 11,9	19,6 \pm 2,4	18,5 \pm 2,3	0,9 \pm 0,2	0,27
14h30-15h00	169,8 \pm 16,8	39,2 \pm 3,6	35,6 \pm 3,3	2,9 \pm 0,4	0,27
15h00-15h30	202,4 \pm 16,8	52,6 \pm 4,1	48,4 \pm 4,0	3,3 \pm 0,4	0,28
15h30-16h00	229,9 \pm 18,0	60,9 \pm 4,2	55,5 \pm 4,0	3,6 \pm 0,4	0,29
16h00-16h30	165,5 \pm 14,4	50,5 \pm 4,0	45,8 \pm 3,7	3,3 \pm 0,4	0,29
16h30-17h00	124,1 \pm 10,9	41,5 \pm 3,7	37,2 \pm 3,6	3,2 \pm 0,4	0,30
17h00-17h30	92,5 \pm 9,2	34,7 \pm 3,5	30,3 \pm 3,2	2,8 \pm 0,3	0,33
17h30-18h00	125,3 \pm 9,5	50,9 \pm 3,7	46,1 \pm 3,6	3,4 \pm 0,4	0,37
18h00-18h30	83,7 \pm 7,8	36,6 \pm 3,1	33,3 \pm 3,0	2,3 \pm 0,4	0,40
18h30-19h00	86,9 \pm 8,1	35,9 \pm 3,1	33,0 \pm 3,0	2,3 \pm 0,3	0,38
19h00-19h30	84,5 \pm 7,3	37,3 \pm 3,2	34,2 \pm 3,0	2,6 \pm 0,4	0,41
19h30-20h00	109,3 \pm 8,4	48,2 \pm 3,7	44,8 \pm 3,6	2,8 \pm 0,4	0,41
20h00-20h30	98,4 \pm 7,0	44,6 \pm 3,3	40,6 \pm 3,3	2,8 \pm 0,4	0,41

Tableau 2.13 – Intervalle de confiance (95%) sur le nombre d’appels composés, le nombre de clients rejoints, le nombre d’appels nécessitant un temps de service, le nombre de “mismatches” et la proportion d’appels rejoints, durant chaque demi-heure d’un jour de semaine.

Ces résultats seront utiles lorsque nous voudrions deviner la politique de composition actuellement utilisée par le centre d’appels. Par exemple, si nous voyons que notre modèle provoque trop de “mismatches” par rapport à la réalité alors nous allons diminuer l’agressivité de son processus de composition.

La proportion des clients rejoints à l’aide de la composition d’appels sortants doit être

Demi-heure i	appels composés	appels rejoints	service requis	“mismatches”	κ_i
10h00-10h30	44,1 \pm 16,5	19,0 \pm 5,0	18,0 \pm 4,6	1,0 \pm 0,5	0,41
10h30-11h00	47,1 \pm 12,3	18,4 \pm 4,3	17,0 \pm 3,9	1,3 \pm 0,7	0,36
11h00-11h30	38,6 \pm 10,9	15,9 \pm 3,7	11,7 \pm 2,7	0,8 \pm 0,4	0,30
11h30-12h00	52,7 \pm 13,4	15,9 \pm 3,7	14,8 \pm 3,4	1,0 \pm 0,4	0,28
12h00-12h30	43,7 \pm 14,3	14,1 \pm 4,2	12,7 \pm 3,6	1,1 \pm 0,6	0,29
12h30-13h00	29,7 \pm 10,9	10,4 \pm 3,6	9,3 \pm 3,3	1,0 \pm 0,5	0,31
13h00-13h30	40,6 \pm 10,9	12,3 \pm 2,9	11,2 \pm 2,7	1,0 \pm 0,4	0,28
13h30-14h00	60,4 \pm 15,5	18,8 \pm 4,4	17,1 \pm 4,0	1,7 \pm 0,5	0,28
14h00-14h30	62,4 \pm 13,6	19,9 \pm 3,9	17,7 \pm 3,4	2,1 \pm 0,7	0,28
14h30-15h00	79,8 \pm 12,3	24,5 \pm 4,1	21,7 \pm 3,8	2,4 \pm 0,7	0,27
15h00-15h30	70,5 \pm 17,3	23,0 \pm 4,4	21,1 \pm 4,1	1,5 \pm 0,4	0,30
15h30-16h00	81,6 \pm 15,0	24,6 \pm 4,3	21,6 \pm 3,9	2,5 \pm 0,7	0,27
16h00-16h30	90,9 \pm 18,4	31,9 \pm 5,5	28,9 \pm 5,0	2,6 \pm 0,7	0,32
16h30-17h00	36,0 \pm 10,3	13,6 \pm 3,7	11,2 \pm 3,1	1,6 \pm 0,6	0,31

Tableau 2.14 – Intervalle de confiance (95%) sur le nombre d’appels composés, le nombre de clients rejoints, le nombre d’appels nécessitant un temps de service, le nombre de “mismatches” et la proportion d’appels rejoints, durant chaque demi-heure d’un samedi.

considérée dans nos modèles. Les raisons pouvant empêcher la connexion pourraient être dues à l’indisponibilité du client, à l’occupation de la ligne téléphonique, etc. À chaque appel composé, nous allons lui associer une probabilité que la communication soit établie. Cette probabilité se base sur les données recueillies. Nous allons la définir par la proportion du nombre d’appels connectés sur le nombre d’appels composés. Une fois de plus, nous allons voir si le temps de la journée a une influence. Intuitivement, il semble qu’un client sera plus facile à rejoindre vers l’heure du souper qu’en milieu d’après-midi.

Les données présentent un nombre de clients rejoints différent de l’addition du nombre de “mismatches” et du nombre d’appels sortants servis. Ceci est dû au fait qu’il y a des communications avec certains clients d’appels sortants qui se terminent brusquement par des erreurs du système. La modélisation du processus de composition ne tiendra pas compte de ces erreurs. Nous allons supposer que les clients rejoints par le centre d’appels sont divisés en deux groupes : soit qu’ils reçoivent le service immédiatement ou ils sont des “mismatches”.

Cette hypothèse modifie la probabilité de rejoindre un client. Le nombre d'appels rejoints devient la somme des "mismatches" et des appels recevant un service. Les tableaux 2.13 et 2.14 présentent les probabilités observées pour chaque demi-heure des jours de semaine et du samedi. Les résultats obtenus nous montrent bien que nous avons raison de supposer que cette probabilité varie durant la journée. En fait, nous avons presque deux fois plus de chances de rejoindre un client en soirée que durant l'après-midi.

2.3 Les préposés

La prochaine section sera consacrée au groupe de préposés dont la tâche est d'offrir un service aux clients du centre d'appels. Jusqu'à présent, nous avons modélisé des aspects se rattachant à ces préposés. Les durées de service pour les appels sortants et entrants en sont des exemples. Afin de compléter notre modélisation d'un centre d'appels, nous devons trouver une manière de représenter le nombre de préposés disponibles en tout temps. L'analyse proposée s'appuie sur une base de données regroupant des informations sur le nombre moyen d'agents en mode entrant et en mode mixte par demi-heure. Nous avons des données sur 253 jours entre le 2 mai 2001 et le 4 mars 2002. Il est important de noter que nous allons éliminer les données d'avant le 6 juin 2001 car le centre n'y traitait pas d'appels sortants.

Il est aussi important de souligner que le nombre d'agents disponibles à chaque demi-heure n'est pas vraiment un aspect à modéliser. Il s'agit plutôt d'une variable de décision. En fait, ce nombre est ce que les gens de Bell veulent qu'on les aide à choisir. Justement, les modèles développés dans ce projet seront utiles pour trouver le nombre d'agents à planifier pour chaque demi-heure afin de respecter une certaine qualité de service. De plus, ils leur permettront d'évaluer la sensibilité du centre d'appels suite à une variation dans le nombre d'agents disponibles.

L'utilité de cette sous-section est d'observer le nombre d'agents planifié sous la politique de gestion actuelle. Cette analyse permettra de calibrer nos modèles pour qu'ils représentent le plus fidèlement la réalité. Ensuite, nous allons pouvoir voir cet aspect comme une variable de décision.

2.3.1 Le nombre de préposés en fonction

La modélisation du nombre de préposés disponibles n'est pas une tâche facile. Nous devons représenter des comportements humains et différents arrêts de travail (pauses, dîners, réunions, etc.). Le centre dispose de deux groupes d'agents : un traitant les appels entrants seulement et un autre fonctionnant en mode mixte. Dans notre cas, nous faisons l'hypothèse que le nombre de préposés disponibles dans chacun des groupes est constant par demi-heure, c'est-à-dire que le nombre peut être modifié seulement à la fin de chaque demi-heure. Ceci permet d'avoir une disponibilité moyenne sans se préoccuper des arrêts de travail. De plus, cette modélisation est pratiquement la seule que nous pouvons proposer, car la base de données présente seulement des observations agrégées par demi-heure.

Dans un premier temps, nous allons supposer que le nombre d'agents disponibles pour une demi-heure spécifique est le même de jour en jour. Le tableau 2.15 présente le nombre moyen d'agents dans les deux groupes pour chaque demi-heure. Nous avons remarqué dans les sections précédentes que le taux d'arrivée des appels dépend de la journée de la semaine. Ainsi, nous pouvons supposer que le centre s'ajuste à cette variation du taux d'arrivée. Il est clair que le centre a besoin de plus d'agents disponibles pour les journées où le trafic d'arrivées est plus fort. Pour cette raison, une deuxième analyse sera proposée et tiendra compte du jour de la semaine. Les tableaux 2.16 et 2.17 présentent le nombre moyen de préposés disponibles en fonction des demi-heures de chaque jour de la semaine. Cette hypothèse semble réaliste, car nous voyons bien que la planification du nombre d'agents est assez différente d'un jour à l'autre.

Dans l'introduction, nous avons dit que chaque agent travaille généralement en mode mixte pour les 3,5 dernières heures de sa journée de travail. Ceci est plus ou moins vrai puisqu'il y a des agents en mode entrant entre 18h00 et 20h30. Il y a deux explications qui nous permettraient de comprendre ce phénomène. Premièrement, il y a des journées où on ne fait aucune composition d'appels sortants. Ceci implique que les agents sont en mode entrant tout au long de la journée. La majorité de ces journées surviennent juste après l'inauguration du centre d'appels. La deuxième explication se base sur la fiabilité du système de composition d'appels sortants. Sans en connaître nécessairement les raisons, il peut se produire une panne du système. Celle-ci entraîne l'arrêt de la composition des appels sortants et oblige tous les agents à se mettre en mode entrant pour le reste de la journée.

Demi-heure	Entrant	Mixte	Demi-heure	Entrant	Mixte
8h00-8h30	11,3	0	14h30-15h00	24,0	12,6
8h30-9h00	18,1	0	15h00-15h30	24,2	17,3
9h00-9h30	23,0	0	15h30-16h00	24,7	20,1
9h30-10h00	25,9	0	16h00-16h30	22,2	18,4
10h00-10h30	26,5	0	16h30-17h00	18,7	16,6
10h30-11h00	27,0	0	17h00-17h30	17,0	12,8
11h00-11h30	26,4	0	17h30-18h00	9,4	19,0
11h30-12h00	26,3	0	18h00-18h30	4,1	19,5
12h00-12h30	23,9	0	18h30-19h00	3,4	18,5
12h30-13h00	24,2	0	19h00-19h30	2,9	17,3
13h00-13h30	30,6	0	19h30-20h00	3,1	19,5
13h30-14h00	29,4	0	20h00-20h30	3,0	18,0
14h00-14h30	25,7	6,1			

Tableau 2.15 – Nombre moyen d’agents disponibles en mode entrant seulement et en mode mixte pour chaque demi-heure des jours de la semaine (lundi, mardi, ..., vendredi).

Nous pourrions rendre la modélisation du nombre d’agents dans chaque groupe plus réaliste en ajoutant un peu de variance. Nous pourrions, par exemple, avoir un facteur représentant l’absentéisme de la journée. Le nombre d’agents disponibles avec ce modèle serait le nombre moyen multiplié par ce facteur. Ce type d’extensions ne fera pas partie de ce mémoire, mais elle pourrait être étudiée dans la suite du projet.

2.3.2 Le taux d’occupation des préposés

Un autre aspect intéressant est le taux d’occupation des agents que l’on observe sous la gestion actuelle du centre d’appels. Le taux d’occupation observé servira de témoin pour ajuster nos modèles. Ensuite, il sera utile pour comparer l’avantage d’utiliser différentes politiques de gestion.

La base de données sur le centre d’appels étudié nous permet de calculer le taux d’occupation observé par demi-heure. Pour chaque demi-heure, nous pouvons calculer la durée de service totale associée aux appels entrants et celle aux appels sortants. La somme de ces

deux durées nous donne le temps total durant lequel les agents sont occupés. Ainsi, le taux d'occupation se calcule en divisant cette somme par la durée totale où les agents étaient disponibles. Voici la formule pour calculer le taux d'occupation pour une demi-heure k :

$$\text{tauxoccup}_k = \frac{\text{TTA}_{e,k} + \text{TTA}_{s,k}}{\text{Disponibilité totale des agents pour la demi-heure } k}, \quad (2.39)$$

où $\text{TTA}_{x,k}$ est la durée totale de service (*Total Time to Answer*) pour les appels de type x dans la demi-heure k . Il est important de noter que l'analyse sur plusieurs demi-heures ne doit pas se faire en prenant la moyenne des taux d'occupation. La bonne méthode est de calculer une moyenne pondérée sur le temps de disponibilité des agents.

Nous allons analyser le taux d'occupation observé en moyenne sur une journée, en moyenne sur chaque demi-heure et en moyenne sur chaque demi-heure de chaque jour de la semaine. Les tableaux 2.18, 2.19 et 2.20 présentent ces analyses.

La première remarque que l'on pourrait faire est que nous observons un faible gain pour les périodes où les appels sortants sont permis, i.e., entre 14h00 et 20h30. Nous croyons que le taux d'occupation des agents devrait être significativement plus élevé lorsque le centre est en mode mixte (après 14h00) que lorsqu'il opère seulement en mode entrant. Ces résultats suggèrent qu'il serait possible d'améliorer la politique de gestion.

Finalement, nous ne pouvons pas dire que le taux d'occupation des agents varie selon le jour de la semaine (à l'exception encore une fois du samedi). Chaque jour présente pratiquement les mêmes comportements. On note, par exemple, un faible taux d'occupation des agents pour les débuts de journée et pour l'heure avant le début de la composition d'appels sortants.

Demi-heure	lundi		mardi		mercredi	
	Entrant	Mixte	Entrant	Mixte	Entrant	Mixte
8h00-8h30	13,0	0,0	11,4	0,0	11,0	0,0
8h30-9h00	20,1	0,0	18,6	0,0	17,8	0,0
9h00-9h30	24,2	0,0	24,3	0,0	22,8	0,0
9h30-10h00	27,5	0,0	27,9	0,0	25,9	0,0
10h00-10h30	29,6	0,0	28,1	0,0	25,9	0,0
10h30-11h00	29,2	0,0	28,1	0,0	26,4	0,0
11h00-11h30	28,5	0,0	27,7	0,0	25,4	0,0
11h30-12h00	27,0	0,0	27,8	0,0	26,5	0,0
12h00-12h30	24,2	0,0	25,8	0,0	23,9	0,0
12h30-13h00	25,0	0,0	25,9	0,0	23,0	0,0
13h00-13h30	30,9	0,0	29,0	0,0	30,0	0,0
13h30-14h00	28,2	0,0	28,9	0,0	29,0	0,0
14h00-14h30	27,5	5,6	26,6	6,1	24,7	6,1
14h30-15h00	26,0	12,4	25,0	14,1	23,2	11,2
15h00-15h30	26,0	18,8	25,6	19,5	24,1	15,9
15h30-16h00	26,0	20,7	26,5	21,7	24,5	19,8
16h00-16h30	22,8	19,0	24,8	20,2	22,8	18,7
16h30-17h00	18,5	16,4	21,4	18,6	19,3	16,9
17h00-17h30	16,8	12,5	19,6	14,1	16,6	12,9
17h30-18h00	10,8	18,8	9,9	21,3	9,7	18,2
18h00-18h30	4,8	19,3	4,1	21,2	4,7	18,9
18h30-19h00	3,8	18,5	3,2	20,6	4,5	17,0
19h00-19h30	3,4	18,6	2,8	19,8	3,7	15,5
19h30-20h00	3,5	20,4	3,3	21,9	3,9	18,2
20h00-20h30	3,5	18,7	3,2	20,8	3,3	17,0

Tableau 2.16 – Nombre moyen d’agents disponibles en mode entrant et mixte pour chaque demi-heure du lundi, mardi et mercredi.

Demi-heure	jeudi		vendredi		samedi	
	Entrant	Mixte	Entrant	Mixte	Entrant	Mixte
8h00-8h30	10,8	0,0	10,2	0,0	0,0	0,0
8h30-9h00	17,3	0,0	17,0	0,0	7,1	0,0
9h00-9h30	22,2	0,0	21,6	0,0	16,4	0,0
9h30-10h00	24,3	0,0	24,4	0,0	14,6	0,0
10h00-10h30	25,3	0,0	23,9	0,0	7,4	7,3
10h30-11h00	26,1	0,0	25,5	0,0	3,4	12,8
11h00-11h30	25,3	0,0	25,4	0,0	2,7	12,9
11h30-12h00	24,7	0,0	25,6	0,0	2,6	14,7
12h00-12h30	23,2	0,0	22,6	0,0	2,0	12,7
12h30-13h00	23,8	0,0	23,3	0,0	1,7	10,9
13h00-13h30	31,7	0,0	31,4	0,0	1,8	10,8
13h30-14h00	30,8	0,0	29,9	0,0	2,1	13,5
14h00-14h30	24,9	6,9	25,0	5,4	2,2	13,7
14h30-15h00	23,9	13,9	21,8	11,4	2,4	13,8
15h00-15h30	24,1	17,4	21,2	15,3	2,6	13,6
15h30-16h00	24,7	19,6	22,2	18,9	2,4	13,2
16h00-16h30	21,5	16,9	19,5	17,1	2,7	15,0
16h30-17h00	17,7	15,8	16,8	15,6	1,7	9,4
17h00-17h30	17,6	12,7	14,6	11,8	0,0	0,0
17h30-18h00	8,9	19,1	7,9	17,4	0,0	0,0
18h00-18h30	3,3	20,5	3,5	17,9	0,0	0,0
18h30-19h00	2,8	19,1	2,8	17,3	0,0	0,0
19h00-19h30	2,5	17,4	2,3	15,6	0,0	0,0
19h30-20h00	2,5	19,7	2,6	17,5	0,0	0,0
20h00-20h30	2,6	18,7	2,5	14,9	0,0	0,0

Tableau 2.17 – Nombre moyen d’agents disponibles en mode entrant et mixte pour chaque demi-heure du jeudi, vendredi et samedi.

Demi-heure	% d'occupation	Demi-heure	% d'occupation
8h00-8h30	58,8	14h30-15h00	73,9
8h30-9h00	64,0	15h00-15h30	73,2
9h00-9h30	70,4	15h30-16h00	74,7
9h30-10h00	72,5	16h00-16h30	78,6
10h00-10h30	73,1	16h30-17h00	80,1
10h30-11h00	73,6	17h00-17h30	73,0
11h00-11h30	75,3	17h30-18h00	76,4
11h30-12h00	73,5	18h00-18h30	78,6
12h00-12h30	72,5	18h30-19h00	76,2
12h30-13h00	70,4	19h00-19h30	75,0
13h00-13h30	59,4	19h30-20h00	72,0
13h30-14h00	60,8	20h00-20h30	69,6
14h00-14h30	69,3	Toutes	72,1

Tableau 2.18 – Taux d'occupation (en pourcentage) des agents en mode entrant et mixte pour chaque demi-heure des jours de la semaine (lundi, mardi, ..., vendredi).

	lundi	mardi	mercredi
Demi-heure	% d'occupation	% d'occupation	% d'occupation
8h00-8h30	60,0	58,7	57,4
8h30-9h00	63,8	64,6	63,1
9h00-9h30	68,7	71,0	68,8
9h30-10h00	71,4	72,2	71,3
10h00-10h30	72,7	72,0	71,3
10h30-11h00	76,1	71,1	71,5
11h00-11h30	74,5	74,9	73,3
11h30-12h00	74,5	71,6	71,4
12h00-12h30	76,2	70,3	71,3
12h30-13h00	74,9	69,3	69,2
13h00-13h30	62,9	60,2	59,9
13h30-14h00	68,6	61,4	58,2
14h00-14h30	70,3	68,3	68,8
14h30-15h00	75,1	72,7	73,9
15h00-15h30	73,0	74,0	71,5
15h30-16h00	75,7	73,4	74,5
16h00-16h30	81,4	77,6	77,9
16h30-17h00	81,8	78,3	81,4
17h00-17h30	71,9	71,4	75,0
17h30-18h00	76,5	76,0	78,4
18h00-18h30	77,2	80,0	80,8
18h30-19h00	78,5	76,3	80,5
19h00-19h30	75,1	76,3	75,8
19h30-20h00	75,6	72,5	68,1
20h00-20h30	69,4	69,8	70,6

Tableau 2.19 – Taux d'occupation (en pourcentage) des agents en mode entrant et mixte pour chaque demi-heure du lundi, mardi et mercredi.

	jeudi	vendredi	samedi
Demi-heure	% d'occupation	% d'occupation	% d'occupation
8h00-8h30	58,5	59,1	—
8h30-9h00	63,4	65,1	55,8
9h00-9h30	71,5	72,1	37,7
9h30-10h00	72,8	74,9	46,3
10h00-10h30	74,1	75,8	75,0
10h30-11h00	72,9	76,2	81,4
11h00-11h30	77,7	76,0	79,8
11h30-12h00	76,2	73,6	75,9
12h00-12h30	72,4	72,2	80,6
12h30-13h00	68,5	69,6	79,2
13h00-13h30	57,5	56,7	79,0
13h30-14h00	58,9	57,7	71,3
14h00-14h30	69,5	69,7	72,3
14h30-15h00	72,8	75,1	72,6
15h00-15h30	72,1	76,2	74,4
15h30-16h00	74,8	75,3	73,4
16h00-16h30	78,2	77,8	71,0
16h30-17h00	80,2	78,8	70,7
17h00-17h30	74,1	72,6	—
17h30-18h00	76,2	74,7	—
18h00-18h30	79,3	75,0	—
18h30-19h00	73,3	72,5	—
19h00-19h30	74,3	73,4	—
19h30-20h00	73,5	70,4	—
20h00-20h30	68,6	69,4	—

Tableau 2.20 – Taux d'occupation (en pourcentage) des agents en mode entrant et mixte pour chaque demi-heure du jeudi, vendredi et samedi.

Chapitre 3

Modèle “réaliste” et expérimentations à l’aide de la simulation

La gestion dans les centres d’appels se base souvent sur des mesures de performance obtenues à l’aide de formules analytiques. La formule d’Erlang-C en est un exemple [19]. Elle est souvent utilisée pour planifier le nombre nécessaire d’agents afin de satisfaire une certaine distribution du temps d’attente des clients. Ces formules ont l’avantage d’estimer *rapidement* plusieurs mesures de performance dont la distribution du temps d’attente, le taux d’occupation des agents, etc. Le problème est que ces formules se basent sur des modèles *simples* ne représentant pas vraiment bien la réalité. Dans [6, 18, 33, 51], les auteurs proposent l’alternative de la simulation. Celle-ci permet l’utilisation d’un modèle plus complexe et plus réaliste, et d’estimer les mêmes mesures qu’avec une formule analytique. Le seul inconvénient est qu’elle ne donne pas une mesure exacte, mais on peut obtenir, avec un grand nombre de répétitions, une valeur assez précise sous forme d’intervalle de confiance.

Le centre d’appels étudié dans ce mémoire est loin d’être conventionnel. Nous avons des appels entrants et sortants, deux groupes d’agents, un processus de composition d’appels sortants, et d’autres particularités. Ceci nous laisse croire qu’il n’est pas approprié d’estimer les mesures de performance de ce centre d’appels à l’aide des formules analytiques connues. La simulation pourrait être une bonne alternative. Klungle [33] décrit les cas où il est

préférable d'utiliser la simulation. Voici ceux qui se rapportent à notre cas :

- Aucun modèle analytique disponible.
- Modèle analytique trop complexe à résoudre.
- Évolution possible du modèle (la simulation permet une adaptation plus facile à des changements dans le centre).
- Modèle demandant la représentation de plusieurs détails.

L'ensemble de ces raisons nous pousse à considérer sérieusement l'utilisation de la simulation. Ce chapitre sera consacré à l'utilisation de la simulation pour faire l'étude de centre d'appels en mode mixte. Tout d'abord, nous allons décrire les étapes suivies pour construire notre modèle le plus réaliste possible. La simulation de ce modèle nous permettra d'obtenir des estimations assez précises sur plusieurs mesures. Ensuite, nous allons voir que la simulation permet une grande variété d'expérimentations. Nous allons l'utiliser pour étudier la sensibilité du centre d'appels sous différents scénarios.

3.1 Modèle “réaliste”

Dans cette section, nous allons présenter la méthode utilisée pour la construction d'un modèle représentant le centre d'appels étudié. Ce modèle a la seule contrainte d'être simulable. Le processus de construction est résumé à la figure 3.1.

Dans un premier temps, nous devons définir les objectifs du modèle. Dans notre cas, ils ont été définis dans l'introduction. En guise de rappel, nous pouvons les résumer en disant que nous voulons développer un modèle réaliste et flexible permettant d'analyser le comportement du centre d'appels sous différentes politiques de gestion. L'étape suivante est la cueillette et l'analyse de données sur le centre d'appels. Tout dépendant des données disponibles, ces analyses fixent les paramètres du modèle. Nous avons déjà consacré le chapitre 2 à cette tâche.

Dans la troisième étape, on tente de construire un modèle à l'aide de l'analyse de ces données et d'informations obtenues sur le centre. Ensuite, nous allons faire quelques expérimentations pour vérifier si l'on doit faire un retour aux étapes de la collecte de données et de conception du modèle. Ce retour sera jugé nécessaire dans le cas où le modèle manque de réalisme. Nous allons faire des changements jusqu'à l'obtention d'un modèle “satisfaisant”, c'est-à-dire celui qui présente des mesures de performance les plus semblables à celles

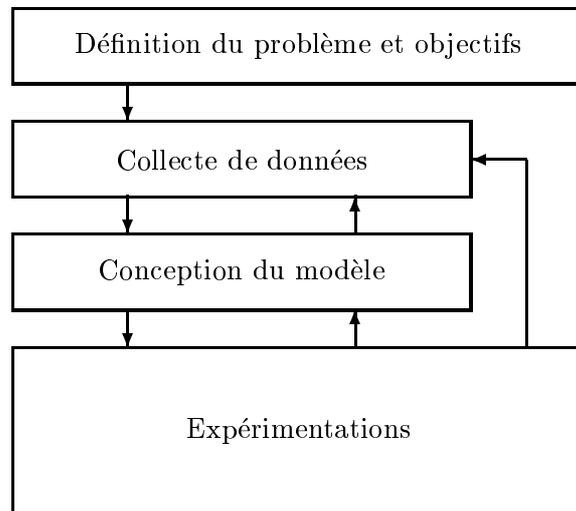


Figure 3.1 – Processus pour la construction d’un modèle de simulation pour un centre d’appels.

que l’on observe dans le vrai centre d’appels. Un modèle “insatisfaisant” est souvent corrigé par une cueillette de données plus précise, mais ceci ne s’applique pas vraiment à notre cas, car les données plus précises ne sont pas disponibles. Nous allons tenter d’améliorer notre modèle en proposant de nouvelles heuristiques sur certains aspects nébuleux du centre d’appels dont le système de composition d’appels sortants, la patience des clients, etc. Les mesures d’intérêt seront les volumes d’appels entrants et sortants servis, le taux d’occupation, le pourcentage d’appels entrants devant attendre moins de 20 secondes dans la file, etc. Pour chacune d’entre elles, nous allons modifier notre modèle afin de minimiser l’écart qui sépare ces mesures simulées sur chaque demi-heure avec celles obtenues avec les données réelles sur le centre.

Les deux premières étapes de la figure 3.1 ont déjà été faites dans les chapitres précédents. L’étape suivante consiste à concevoir un modèle mathématique représentant le centre d’appels. Son but principal est d’être simulable afin d’estimer ses mesures de performance. Cette nécessité nous a amené à construire une librairie logicielle pour la simulation d’un tel modèle mathématique. Nous avons divisé notre logiciel en modules dont la structure s’inspire de la figure 1.1 : on consacre un module à chacune des boîtes du diagramme et les flèches entre les boîtes représentent les interrelations qui sont permises entre les modules.

La division du modèle mathématique en modules est très pratique. Elle permet d’avoir

une bonne compréhension de la structure et des relations entre les différents aspects du modèle en faisant abstraction de plusieurs détails. De plus, elle permet une stabilité dans la structure générale du modèle même lorsque celui-ci évolue. La modification de certains détails se fait à l'intérieur du module correspondant et ne concerne pas les autres modules.

Notre logiciel doit utiliser plusieurs outils de simulation dont une liste d'événements, des générateurs pseudo-aléatoires, des outils statistiques, etc. Au lieu de programmer ces outils, nous avons décidé d'utiliser une librairie de simulation générale qui s'applique bien à nos besoins. Il existe un bon choix de logiciels disponibles. Le plus connu est sûrement Arena qui permet une construction graphique. Ce logiciel commercial a plusieurs désavantages. Premièrement, il n'est pas gratuit et n'est pas reconnu pour faire des simulations rapides. Ensuite, son interface graphique est pratique pour la simulation de modèles conventionnels, mais son utilisation devient lourde pour les cas spéciaux. Notre centre d'appels a plusieurs particularités qui pourraient alourdir l'utilisation de ce logiciel. En plus, la construction graphique de modèle n'est pas vraiment un avantage puisque la librairie sera utilisée par des gens ayant des bonnes connaissances en programmation. Ces raisons nous poussent à considérer d'autres alternatives.

Dans [41], les auteurs énumèrent quelques logiciels répondant à nos besoins. Ceux-ci utilisent une grande diversité de langages de programmation : C, FORTRAN, Modula-2, Java, etc. Leurs avantages et désavantages proviennent la plupart du temps du langage de programmation utilisé. P. L'Ecuyer, professeur à l'Université de Montréal, et son équipe ont construit différentes bibliothèques d'outils pour la simulation en général. Ces logiciels sont disponibles en 3 langages de programmation : SIMOD en Modula-2 [38], SSC en C [40] et SSJ en Java [39]. Cette option est intéressante pour plusieurs raisons. Premièrement, les bibliothèques sont "open-source" et gratuites. Ceci permet l'ajout d'une couche logicielle pouvant traiter plus particulièrement les modèles de centres d'appels. Le choix de la version à utiliser dans le cadre de ce mémoire ne fut pas difficile. Tout d'abord, nous avons éliminé SIMOD puisque le langage Modula-2 n'est plus supporté. Ensuite, nous avons éliminé la version Java malgré son avantage de permettre une programmation-objet mieux adaptée aux centres d'appels. Nous avons eu peur que SSJ soit trop lent pour la simulation de nos modèles. Nous avons alors choisi SSC comme base à notre bibliothèque de simulation de centre d'appels et ainsi, elle est entièrement programmée en C. Lors de son développement et de son utilisation, nous avons mis en doute notre choix. Dans un premier temps, nous avons

réalisé qu’une bonne programmation-objet aurait aidé à la clarté et à la compréhension de notre librairie. La division du modèle mathématique en modules suggère l’utilisation d’une programmation-objet. Nous nous sommes rendus compte que le langage C complexifie l’utilisation et le développement d’un logiciel modulaire. Nous nous sommes même demandés si la rapidité du langage C était futile par rapport au gain que l’on ferait en clarté avec l’utilisation du langage Java. De plus, l’article [41] montre que la rapidité de SSJ n’est pas si mauvaise par rapport à SSC. Ces raisons nous laissent croire qu’une implantation du logiciel pour la simulation de centre d’appels en Java serait peut-être souhaitable. Il en demeure que notre version en C est suffisante pour l’instant. Cette proposition de traduction et de réorganisation de notre librairie est une suggestion qui pourrait faire partie d’un futur projet de maîtrise.

Nous allons maintenant présenter les différents modules de la librairie. Ceux-ci donnent la structure du modèle. Les modules Incall, Outcall, Waitqueue, Dialer et Servers représentent respectivement les appels entrants, les appels sortants, la file d’attente, le système de composition d’appels sortants et les agents. Le logiciel ajoute les modules Distribution et Statistic qui permettent de générer des variables aléatoires caractérisées par des lois ou des processus stochastiques non conventionnels et de cumuler des statistiques journalières sur des mesures de performance du modèle. Le développement des modules s’est fait à l’aide des analyses présentées au chapitre 2 et de différentes informations obtenues par l’entremise d’employés travaillant pour Bell.

3.1.1 Construction d’un modèle représentant le centre d’appels

Dans cette sous-section, nous allons décrire un premier modèle construit à l’aide de notre librairie de simulation. Nous allons définir le comportement des modules et les interactions possibles entre eux. Des détails plus spécifiques sur la librairie se retrouvent dans son guide d’utilisation en annexe.

Module “Incall”

Ce module représente les aspects reliés aux appels entrants. On y modélise le processus stochastique des arrivées d’appels et le temps maximal d’attente accepté (A_e) des clients entrants. On calcule aussi des statistiques sur le volume journalier et par demi-heure d’appels

entrants arrivées, servis, perdus, ayant attendu plus de 20 secondes, etc.

Le processus d'arrivée d'appels utilisé est celui qui semble le plus près de la réalité parmi ceux étudiés à la sous-section 2.1.1. Ce dernier est le processus de Whitt avec un facteur W définie par une loi gamma. En guise de rappel, il s'agit d'un processus de Poisson qui définit, pour chaque journée, un facteur aléatoire W multipliant une fonction déterministe de taux d'arrivée $\lambda(t) = \lambda_i$ lorsque t est dans la demi-heure i . Ce processus semble assez riche et ses hypothèses semblent logiques.

La simulation de ce genre de processus est assez simple. Au début d'une journée, on génère une variable W suivant la loi gamma définie par les paramètres $\hat{\alpha}'$ et $\hat{\beta}'$ estimés dans la sous-section 2.1.1. L'algorithme utilisé pour générer une telle variable aléatoire est celle du module Rand1 de la librairie de simulation SSC. Celui-ci utilise la méthode d'acceptation-rejet décrite dans [14]. Soit w la valeur prise par W , nous allons maintenant supposer que les arrivées suivent un processus de Poisson non homogène de taux $w\lambda(t)$. Nous allons répéter ces étapes au début de chaque nouvelle journée. La méthode utilisée pour générer le processus de Poisson non homogène est l'inversion et elle est décrite dans [37]. Nous allons toutefois en faire le rappel et l'appliquer au cas d'une fonction de taux d'arrivée linéaire par morceaux. Supposons que nous sommes à l'instant t_{i-1} , que c est l'instant du prochain changement de taux d'arrivée. Voici la méthode pour générer l'instant t_i du prochain appel :

1. Posons $t = t_{i-1}$
2. Génère X comme une exponentielle de taux $w\lambda(t)$
3. $t_i = t + X$
4. Tant que $t_i > c$
 - (a) $X = c - t + (t_i - c)(w\lambda(t))/(w\lambda(c))$
 - (b) $t_i = t + X$
 - (c) si $t_i > c$: $c =$ prochain instant de changement de taux
5. Retourne t_i

La fonction pour générer une exponentielle est l'inversion (p.460 [37]). Dans nos simulations, les instants d'arrivée t_1, t_2, \dots sont générés les uns à la suite des autres à l'aide de cet algorithme. Dès qu'un appel est arrivé, on planifie l'arrivée du prochain. Nous n'avons pas à tenir compte des heures d'ouverture du centre d'appels, car la fonction de taux s'en charge.

Le deuxième aspect à simuler est le temps maximal d'attente accepté. La sous-section

2.1.3 ne traite pas de cet aspect puisqu'il n'existe pas de données pertinentes sur le sujet. Quelques expériences de simulation nous ont permis d'observer des mesures de performance assez réalistes en supposant la variable A_e de loi exponentielle mixte. Dans 0,5 % des cas nous utilisons une loi exponentielle avec une moyenne égale à 3 secondes et dans 99,5 % des cas nous utilisons la loi exponentielle avec une moyenne variant dans le temps : égale à 400 secondes pour la période 8h00-9h30, 700 secondes pour la période 9h30-10h30, 600 secondes pour la période 10h30-13h00, 500 secondes pour la période 13h00-19h30, 100 secondes pour la période 19h30-20h00 et 50 secondes pour la période 20h00-20h30. Cette combinaison de loi exponentielle nous permet de représenter une faible proportion de clients très peu patients. Nous avons observé la présence de ce type de clients dans les données sur la distribution de l'attente des abandons (voir sous-section 2.1.3). Les moyennes pour le deuxième type de client ont été choisies afin de minimiser la distance sur chaque demi-heure entre le nombre d'abandons simulé et celui observé dans les données sur le centre d'appels. Cette méthode ne garantit pas une modélisation réaliste, mais c'est le mieux que nous puissions faire. Nous allons avoir besoin de générer cette variable uniquement dans le cas où un client doit être inséré dans la file d'attente. La génération se fera à l'aide de la méthode d'inversion pour une variable exponentielle (p.460 [37]). Lorsqu'un serveur tentera de servir un client provenant de la file d'attente, nous devons vérifier s'il a attendu plus longtemps que son A_e . Si c'est le cas alors nous avons un nouvel abandon et l'agent tente de servir un autre client. Sinon, le client quitte la file et débute son service.

Ce module peut interagir avec les autres modules. Ceux-ci peuvent le questionner sur des statistiques concernant les appels entrants : distribution sur le temps d'attente, nombre d'abandons, etc. Ils peuvent aussi obtenir un indice de qualité de service sur les appels entrants. Dans notre cas, cet indice est le pourcentage d'appels entrants ayant attendu plus de 20 secondes au cours des 10 dernières minutes. De plus, ce module doit communiquer avec le module des serveurs pour l'informer de chaque arrivée d'appel entrant.

Module “Servers”

Les groupes d'agents dans le centre d'appels sont représentés par ce module. Dans l'introduction, nous avons décrit qu'il y avait deux groupes de préposés. Les préposés du premier groupe ne servent que les appels entrants tandis que ceux du deuxième servent autant les appels entrants que sortants. Afin de simplifier l'étude de ces deux groupes, nous allons

dénoter le premier groupe par la lettre I et le deuxième par la lettre B. La division en deux groupes ne doit pas être négligée, car la simulation d'un modèle avec un seul groupe en mode mixte donne des résultats erronés. Dans le rapport [7], nous avons eu l'information que tous les agents étaient de type B après 14h00. La simulation de ce modèle nous donnait un taux d'occupation des agents et un volume d'appels sortants trop élevés par rapport aux valeurs observées dans la réalité. En supposant que tous les agents pouvaient traiter les deux types d'appels, nous obtenions une meilleure performance du centre d'appels. Nous avons obtenu des mesures plus réalistes dès que nous avons corrigé le modèle.

Il est important de noter que le nombre d'agents dans chaque groupe est une variable de décision, i.e. Bell veut qu'on les aide à trouver le nombre d'agents à planifier dans chaque groupe. Pour l'instant, nous allons utiliser le nombre d'agents disponibles calculé dans la sous-section 2.3.1. L'utilisation de cette disponibilité d'agents nous permettra de développer un modèle qui présente des mesures de performance réalistes avec les variables de décision actuelles. Par la suite, nous allons pouvoir voir le nombre d'agents dans les deux groupes comme une variable de décision.

Notre modèle mathématique fait l'hypothèse que nous avons un nombre d'agents dans chaque groupe qui est constant par demi-heure. Cette hypothèse permet d'obtenir une moyenne du nombre d'agents au travail sans se préoccuper des pauses-café, des séances de formation, des dîners, etc. Nous avons éliminé ces détails, car nous aurions des simulations plus lentes, et les données recueillies ne nous permettent pas de modéliser ce genre de détails. Au cours d'une simulation, ce module fournit le nombre d'agents disponibles, occupés et libres dans chaque groupe.

En plus de modéliser le nombre d'agents disponibles, ce module s'occupe de la durée de service pour chaque type d'appels. Dans les sous-sections 2.1.2 et 2.2.1, nous avons tenté de modéliser respectivement la durée de service pour les appels entrants et celle pour les appels sortants. La sous-section 2.1.2 mentionne que l'analyse sur les appels entrants se base sur l'occupation des agents avec ce type d'appels. Cette information biaise quelque peu l'analyse sur les durées de service individuelles. Afin de réduire ce biais, nous avons estimé les paramètres d'une loi gamma sur des données agrégées pour quelques demi-heures semblables. Le tableau 3.1 présente les paramètres estimés de la loi gamma que nous allons utiliser dans notre modèle. Le choix de la loi gamma vient du fait que nous n'avons pas réussi à rejeter indirectement cette hypothèse. Encore une fois, nous utilisons l'algorithme

Demi-heure	$\hat{\alpha}$	$\hat{\beta}$
8h00-12h00	0,729	817,0
12h00-14h30	0,620	927,6
14h30-16h30	0,755	753,8
16h30-18h30	0,553	996,9
18h30-20h30	0,510	1018,6

Tableau 3.1 – Paramètres de la loi gamma représentant les durées de service d’appels entrants.

de SSC pour générer une telle variable aléatoire.

Dans la sous-section 2.2.1, nous avons eu de la difficulté à ajuster une loi sur les durées d’appels sortants empiriques. En fait, nous n’avons pas réussi à obtenir une modélisation que nous ne rejetions pas. Nous aurions pu utiliser la loi Weibull avec les paramètres calculés dans la sous-section 2.2.1 puisqu’elle présente la plus petite différence entre la fonction de distribution estimée et empirique. Au lieu de se contenter de la moins pire, nous allons utiliser l’avantage d’avoir un grand nombre d’observations et générer les durées individuelles à partir d’une fonction de répartition empirique construite par noyaux. Nous avons déjà décrit l’idée derrière cette méthode à la fin de la sous-section 2.2.1. La librairie UNURAN nous permet d’inclure cette méthode dans notre librairie de simulation. Des informations supplémentaires sont disponibles dans le guide d’utilisation de la librairie UNURAN [42].

Nous allons maintenant décrire les interactions possibles avec les autres modules. Tout d’abord, le service d’un appel entrant ou sortant nécessite toujours l’intervention de ce module. Pour débiter le service d’un appel, il faut qu’il y ait au moins un agent de libre pouvant servir ce type d’appel. Dans le cas où il n’y a pas d’agent inoccupé, nous faisons appel au module de la file d’attente pour qu’il puisse traiter son attente. Dans le cas contraire, nous pouvons servir l’appel et planifier sa fin du service selon la loi associée à sa durée de service et l’instant courant. À la fin du service, nous libérons l’agent et on tente de lui associer un appel en attente. Ceci implique que le groupe d’agents doit s’informer sur la possibilité de servir un client en attente. Ensuite, s’il reste des agents inoccupés alors on demande au système de composition d’appels sortants de décider du nombre d’appels sortants supplémentaires à composer. L’algorithme de décision du système de composition se trouve dans le module Dialer. Nous faisons l’hypothèse que le système de composition

prend une décision à chaque fin de service d'un appel.

Les autres modules ont quelquefois besoin de connaître l'état des groupes de préposés. Nous définissons cet état par le nombre d'agents disponibles et libres dans chaque groupe. Il se peut, par exemple, que l'algorithme de décision pour le système de composition d'appels sortants ait besoin de cette information. En tout temps, il est possible de connaître l'état en questionnant ce module.

Finalement, le module se charge aussi de faire une cueillette de statistiques sur les agents. Celles-ci nous permettent de connaître le taux d'occupation des agents, le volume d'appels entrants servis, le volumes d'appels sortants servis, etc.

Module "Waitqueue"

Les détails concernant la file d'attente sont modélisés dans ce module. Nous faisons l'hypothèse que la file d'attente est de capacité infinie. La règle de priorité pour les appels entrants est du type premier arrivé - premier servi (FIFO : Firts In First Out). Il n'y a qu'une seule file pour les appels entrants et sortants, mais les appels sortants ont priorité sur les appels entrants.

Pour chaque arrivée d'appel, on demande aux groupes de serveurs de débiter son service immédiatement. Si ce n'est pas possible alors on insère cet appel dans la file d'attente : à la fin pour un appel entrant et après le dernier appel sortant pour un appel sortant. Au moment de l'insertion, on lui associe un temps maximal d'attente accepté A qui est calculé par le module correspondant à son type d'appel. La file d'attente se libère lorsque nous avons un abandon ou lorsque les groupes d'agents demandent à servir des appels en attente. Dans le dernier cas, on donne aux agents le premier appel valide dans la file dont le temps d'attente réel n'a pas dépassé son A . À chaque fois que l'attente d'un client dépasse son A , nous avons un nouvel abandon.

En plus de gérer les relations des autres modules avec la file d'attente, ce module permet de calculer plusieurs statistiques. Nous calculons le temps d'attente moyen par appel, la taille moyenne de la file, etc. Ces mesures seront utiles pour estimer l'attente générée sous différents scénarios.

Module “Dialer”

Dans l’introduction, nous avons discuté du problème associé à la modélisation du système de composition des appels sortants. En guise de rappel, nous avons noté qu’il y a très peu d’informations disponibles sur l’algorithme de décision développé par la compagnie Avaya. Suite à quelques discussions avec des gens ayant suivi une formation sur le système, nous savons qu’il se base, entre autres, sur le nombre d’agents libres et sur une certaine mesure de la qualité de service. Dans notre modèle, nous pouvons avoir accès au nombre d’agents libres en questionnant le module Servers et nous pouvons avoir l’information sur une mesure de qualité de service en questionnant les différents modules. Nous avons eu l’information que le système de composition réel considère le pourcentage d’appels entrants ayant attendu plus de 20 secondes au cours des 10 dernières minutes. Ce pourcentage est calculé en temps réel par un système nommé MAX. Il est possible de reproduire cette information dans notre modèle, mais le temps de mise-à-jour ralentit énormément la vitesse de nos simulations. Nous avons décidé d’agréger par minute les informations nécessaires au calcul de cette qualité de service afin d’accélérer la simulation. Ainsi, le calcul se fait en observant les 10 dernières minutes. Cette heuristique permet d’avoir un indice de la qualité de service fiable et rapide à obtenir. Le problème est que nous n’avons aucune idée comment le vrai système utilise cette valeur. Au cours d’une simulation, cette valeur est fournie par le module Incall.

Les rapports [7, 8, 21, 20] présentent une petite partie des algorithmes de décisions que nous avons testés. Différentes expérimentations nous ont permis de développer un algorithme de décisions qui semble produire un comportement assez près de celui observé dans la réalité. Le critère de réalisme se base sur une comparaison entre les mesures de performance simulées et celles obtenues avec les données sur le centre d’appels réel. Nous avons retenu l’algorithme qui présentait le plus faible écart.

Nous allons maintenant décrire l’algorithme utilisé dans notre modèle. À chaque fin du service d’un client, nous décidons du nombre d’appels sortants supplémentaires à composer. Tout d’abord, on vérifie les trois conditions suivantes : si le nombre d’agents libres dans le groupe B est supérieur à 2, si le nombre total d’agents libres dans les deux groupes est supérieur à 4 et si plus de 75 % des appels entrants depuis les 10 dernières minutes ont attendu moins de 20 secondes. Si ces trois conditions ne sont pas toutes satisfaites, on n’effectue aucune composition d’appel. Dans le cas contraire, le nombre d’appels à composer

est égal à deux fois le nombre d’agents libres dans le groupe B si le pourcentage de “mismatches” depuis les 15 dernières minutes ne dépasse pas 8% et à seulement une fois le nombre d’agents libres dans le groupe B sinon. Il est important de noter que nous ne supposons pas que cet algorithme est optimal. Le but de cet exercice est d’obtenir un modèle qui semble réaliste par rapport à ce que nous observons présentement dans le centre d’appels. Nous allons présenter des améliorations possibles à ce processus de composition dans la suite du mémoire. En plus d’avoir à modéliser un système de composition inconnu, notre modèle doit tenir compte d’un facteur humain qui représente les décisions prises par les gestionnaires d’accès. Ces derniers peuvent avoir une grande influence sur la performance du centre d’appels. Le processus de composition se termine 5 minutes avant la fin de la journée. Il est inutile d’appeler des clients en sachant que le centre d’appels ferme bientôt et c’est la règle présentement utilisée dans le centre.

Pour chaque appel composé, on doit modéliser si le client a pu être rejoint. Nous avons analysé cet aspect dans le chapitre 2. La sous-section 2.2.2 présente les probabilités de connexion pour les appels sortants qui dépendent de l’heure de la journée. Notre modèle utilise exactement ces probabilités. Pour chaque appel composé à l’instant t , nous allons générer une variable Z suivant une Bernoulli de probabilité κ_i lorsque t est dans la demi-heure i . Il suffit de générer U une variable uniforme(0,1) et de fixer la valeur 1 à Z si $U < \kappa_i$ et la valeur 0 sinon. La valeur $Z = 1$ est associée à une connexion réussie.

Finalement, le module recueille quelques statistiques journalières sur le système de composition. Le nombre d’appels composés, le nombre de clients rejoints et le nombre de “mismatches” sont les principales statistiques d’intérêt. À tout moment, le module peut communiquer ces informations aux autres modules.

Module “Outcall”

Ce module gère les aspects reliés aux clients d’appels sortants. On y modélise le temps maximal d’attente accepté A_S dans la file d’attente. Le modèle suppose que le client d’un appel sortant est prêt à attendre 5 secondes au maximum et il raccroche ensuite. Cette durée pourrait être associée à un message enregistré que le centre fait écouter au client rejoint. En d’autres mots, le client écoute le message et ensuite, il raccroche la ligne si son service ne peut pas débuter immédiatement. Cette hypothèse semble logique, car il y a très peu de

chance qu'une personne recevant un appel ait la patience d'attendre. Cette affirmation ne se base sur aucune donnée, car nous n'avons pas accès à cette information.

Ce module interagit avec les autres modules de la librairie. Le module du système de composition informe ce module à chaque fois qu'un nouvel appel sortant a été rejoint. Ensuite, il avise le groupe de serveurs qu'il y a un nouvel appel sortant à servir.

Tout au long de la simulation, ce module recueille des mesures sur les appels sortants dont le nombre d'appels servis et la distribution de son attente. Il est possible d'obtenir ces statistiques en questionnant le module.

Autres modules

Notre librairie de simulation comprend trois autres modules qui n'influencent pas la structure du modèle. Un premier qui permet de générer des variables aléatoires suivant une loi ou un processus stochastique non conventionnel. Un autre qui récolte des mesures journalières sur le modèle simulé et qui permet de présenter un rapport sur celles-ci (sous forme d'intervalles de confiance). Finalement, le dernier module initialise les modules à chaque début de journée et gère la simulation du modèle sur plusieurs journées. Des informations plus détaillées sont disponibles dans le guide d'utilisation fourni en annexe.

3.1.2 Simulation du modèle

Nous allons maintenant décrire le déroulement de la simulation d'une journée. Au début de chaque journée, on initialise tous les modules de la librairie de simulation. Ensuite, on planifie la première arrivée à partir de 7h55 et on débute la journée à 8h00 avec un centre d'appels vide. Nous laissons les modules interagir entre eux jusqu'à 14h00, l'instant où nous activons le système de composition des appels sortants. Ceci signifie que le module Dialer devient actif après 14h00. Ensuite, nous arrêtons les arrivées des appels entrants à 20h30 et la composition d'appels sortants à 20h25. La simulation d'une journée se termine lorsque le dernier client restant dans le centre d'appels a été servi. Au courant de la simulation de plusieurs journées, on cumule des mesures journalières et par demi-heure sur différents aspects du centre. À la fin, nous présentons un résumé de ces mesures en supposant que les journées simulées sont indépendantes entre elles.

Lorsqu'on simule le modèle, on remarque des performances peu réalistes au niveau de la qualité de service et du taux d'occupation. Dans la période où le centre traite uniquement des appels entrants, on remarque une trop bonne qualité de service accompagnée d'un faible taux d'occupation des agents. Ces observations pourraient être expliquées par un trop grand nombre d'agents disponibles. Les agents dans notre modèle sont peut-être un peu trop "efficaces", c'est-à-dire qu'il y a des pertes de temps que nous ne modélisons pas bien. Pour tester cette hypothèse, nous avons supposé que les agents du groupe I sont disponibles uniquement durant 90 % du temps. Nous pouvons facilement modéliser ce détail en diminuant le nombre d'agents disponibles de 10 %. Cette modification améliore grandement le réalisme du modèle. Par ailleurs, nous avons appliqué cette notion de perte de temps aux agents du groupe B, car le volume d'appels sortants servis était trop élevé. Pour ce groupe, nous avons supposé qu'un agent est disponible durant 85 % du temps.

Nous allons maintenant présenter les analyses statistiques sur lesquelles nous nous basons pour dire que ce modèle semble réaliste. Nous comparons d'abord les mesures journalières observées dans la réalité avec celles obtenues en simulant le modèle. On s'intéresse principalement au nombre d'arrivées d'appels entrants, au nombre d'abandons, au nombre d'appels sortants composés, au nombre d'appels sortants servis, au nombre de "mismatches", au taux d'occupation des agents ainsi qu'au pourcentage d'appels entrants ayant attendu moins de 20 secondes (souvent référé comme la qualité de service). Le tableau 3.2 présente les mesures obtenues par la simulation du modèle (première colonne) et celles obtenues par l'analyse des données recueillies dans le centre d'appels étudié (deuxième colonne). Les valeurs de la première colonne ont été calculées suite à la simulation de 10000 jours. Nous présentons ces résultats sous la forme d'intervalles de confiance à 95% en supposant que nous avons 10000 jours indépendants.

Les résultats du tableau 3.2 montrent que le modèle proposé fournit des mesures proches de celles que l'on observe dans la réalité : nous observons que les intervalles de confiance des mesures simulées sont incluses dans celles des mesures empiriques. Ainsi, nous n'avons pas une différence significative et nous ne pouvons pas rejeter statistiquement l'hypothèse selon laquelle notre modèle représente bien la réalité pour ces 7 mesures journalières. Cependant, ceci est peu révélateur du comportement de notre modèle au courant de la journée. Nous avons trouvé une manière de l'évaluer en calculant sur chaque demi-heure, le volume moyen d'appels entrants servis, le volume moyen d'appels sortants servis, le taux d'occupation des

	Mesures simulées	Mesures empiriques
Appels entrants reçus par jour	1200,8 ± 4,5 appels	1201,9 ± 32,1 appels
Abandons par jour	31,8 ± 0,8 abandons	31,9 ± 3,1 abandons
Appels sortants composés par jour	1666,0 ± 5,5 appels	1664,0 ± 88,5 appels
Appels sortants servis par jour	511,9 ± 1,7 appels	503,9 ± 26,3 appels
Nombre de “mismatches” par jour	37,6 ± 0,2 “mismatches”	35,2 ± 0,2 “mismatches”
Occupation des agents (%)	71,5 ± 0,1 %	72,3 ± 1,1 %
Qualité de service	86,3 ± 0,1 %	86,1 ± 1,2 %

Tableau 3.2 – Tableau comparatif entre les mesures journalières obtenues par la simulation de notre modèle mathématique et celles observées dans la réalité. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

agents, le nombre moyen d’abandons et le pourcentage de clients ayant attendu moins de 20 secondes. Le volume d’appels pour une demi-heure est défini comme le temps de service consacré à un type d’appels (entrants ou sortants) par les agents disponibles dans cette demi-heure. Ensuite, le taux d’occupation d’une demi-heure est la proportion du temps qu’un agent consacre à servir un client, i.e. la somme des volumes d’appels entrants et sortants servis divisée par la durée totale de la disponibilité des agents. Pour une demi-heure donnée, le nombre d’abandons et le pourcentage de clients entrants ayant attendu moins de 20 secondes sont calculés à partir des appels entrants qui sont arrivés dans cette demi-heure. La base de données disponibles nous permet d’obtenir la valeur empirique sur chaque demi-heure de ces 5 mesures. La comparaison des mesures se basera sur la différence relative entre les valeurs simulées et empiriques pour chaque demi-heure. Soient E la valeur empirique et F la valeur simulée pour une demi-heure. Nous allons définir la différence relative en pourcentage :

$$D\% = 100 * (F - E)/E. \quad (3.1)$$

Nous allons calculer cette différence relative pour les volumes d’appels servis et le nombre d’abandons. Pour les deux autres mesures, nous allons simplement prendre la différence entre la valeur obtenue par simulation et celle calculée de manière empirique. Pour l’instant, nous allons supposer qu’une différence relative comprise entre -10% et 10% est acceptable.

Les différences obtenues pour chacun des aspects sont présentées dans les tableaux 3.2 à 3.6 et nous montrent que nous avons une différence acceptable pour presque tous les cas.

En fait, il n'y a que le volume d'appels sortants servis de l'avant-dernière demi-heure et le volume d'appels entrants servis de 2 autres demi-heures qui font défaut. Pour l'écart concernant le volume d'appels sortants, il est possible que notre système de composition d'appels soit trop agressif pour cette demi-heure. Il se peut que le système réel modifie son algorithme en sachant que la journée se termine bientôt. Au niveau du volume d'appels entrants servis, il n'est pas évident de proposer une explication logique à ce que nous observons. Nous pouvons peut-être l'expliquer en partie par les difficultés rencontrées dans la modélisation de la durée de service des appels entrants (voir sous-section 2.1.2).

Bien que nous ayons des différences acceptables dans la majorité des cas, nous observons une importante différence au niveau de la qualité de service des appels entrants pour la période entre 10h30 et 12h30 (demi-heures 6 à 9). Le pourcentage des clients ayant attendu plus de 20 secondes (figure 3.4) et la quantité d'abandons (figure 3.6) sont trop élevés. Il y a deux raisons qui pourraient expliquer ce résultat. Il se peut que le processus d'arrivée des appels entrants soit trop fortement corrélé : nous allons voir plus tard que c'est effectivement le cas. Cette période de la journée est associée aux plus hauts taux d'arrivée et ainsi, un facteur W élevé engorge le centre et entraîne une mauvaise qualité de service. Ceci influence considérablement la qualité de service globale d'une demi-heure, car un faible facteur W ne peut pas avoir la même influence : le pourcentage de clients étant servis dans les 20 secondes est rapidement borné à 100 %. La deuxième raison est en relation avec la première. Dans les données, nous observons des fortes intensités de trafic, mais ces dernières ne sont pas nécessairement associées à des mauvaises qualités de service puisque le nombre d'agents disponibles peut varier et être suffisant. Le problème avec notre modèle est que nous avons un nombre fixe d'agents disponibles pour tous les jours et toutes les intensités de trafic. Nous pouvons peut-être réduire ce problème en étudiant les journées séparément : le nombre d'agents disponibles sera mieux ajusté aux intensités de trafic rencontrées dans la journée. Cette division par jour sera expérimentée dans la suite de cette sous-section. Pour ce qui est du processus d'arrivée fortement corrélé, la suite du projet étudiera la possibilité d'utiliser un processus d'arrivée Bayésien qui permettrait des corrélations plus douces [47].

Le tableau et les figures montrent que nous avons un modèle qui présente un comportement assez réaliste. Il reste à savoir si une différence relative comprise entre -10% et 10% est réellement acceptable. Pour ce faire, nous avons calculé des p -valeurs associées aux mesures de performance obtenues sur chaque demi-heure, à l'aide de la méthode de "boots-

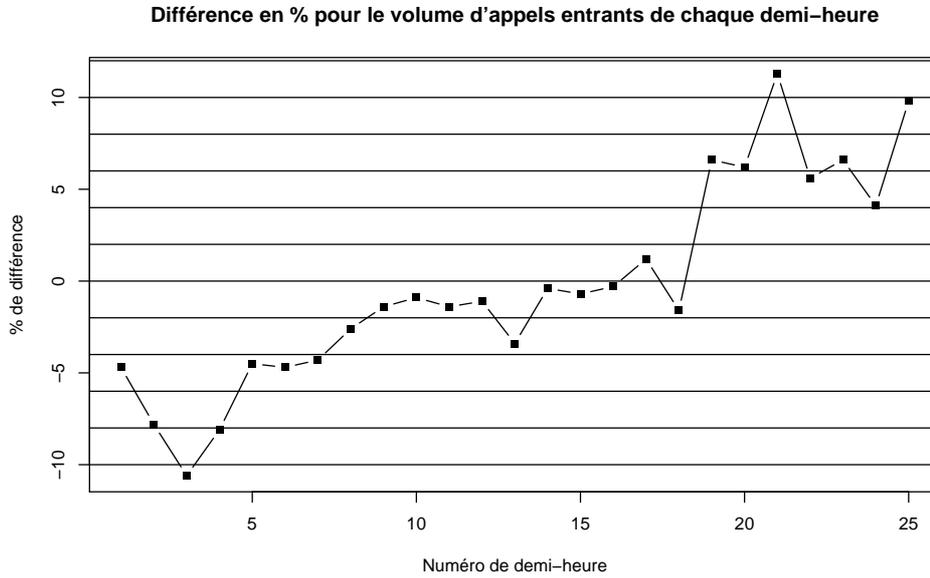


Figure 3.2 – Différence relative entre la valeur empirique et simulée du volume moyen d'appels entrants servis pour chaque demi-heure.

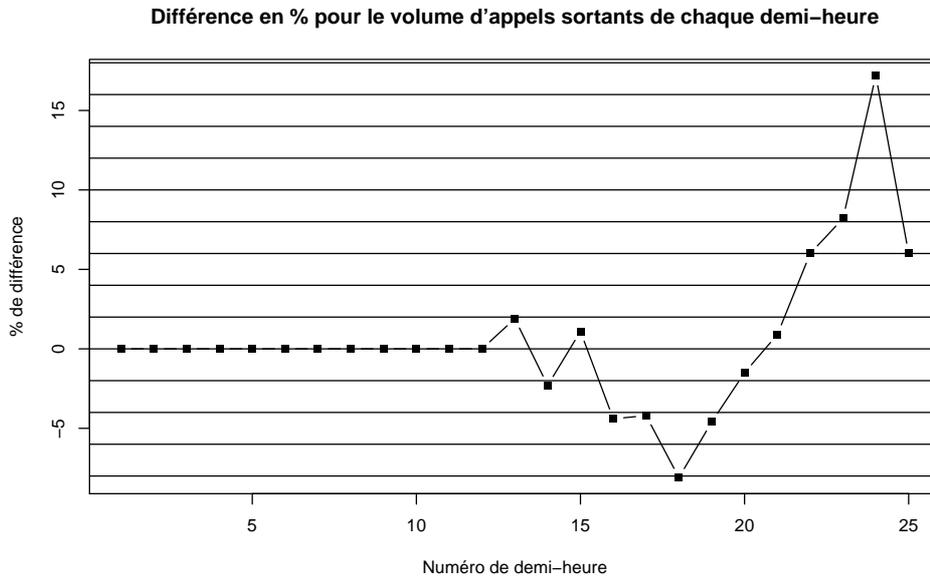


Figure 3.3 – Différence relative entre la valeur empirique et simulée du volume moyen d'appels sortants servis pour chaque demi-heure.

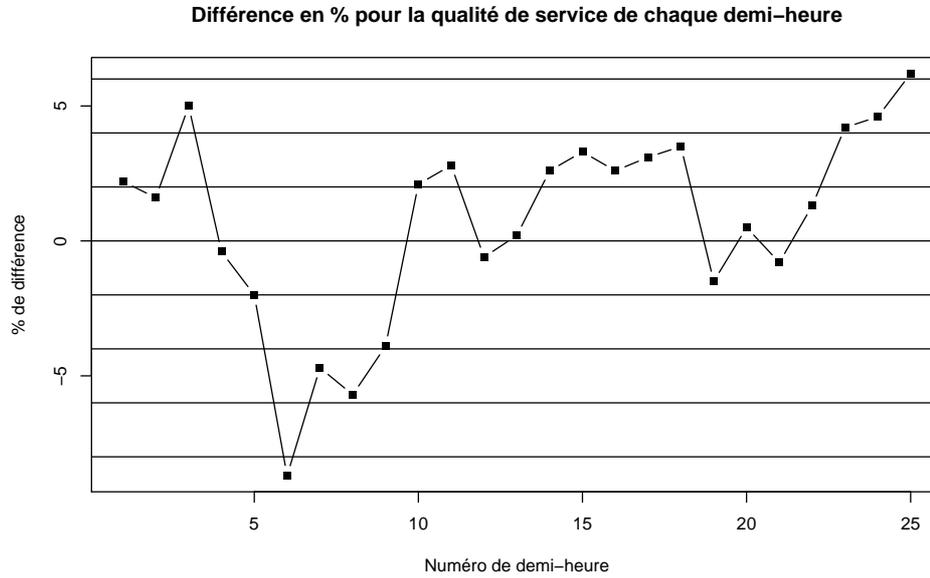


Figure 3.4 – Différence entre le pourcentage empirique et simulé de la qualité de service observée pour chaque demi-heure.

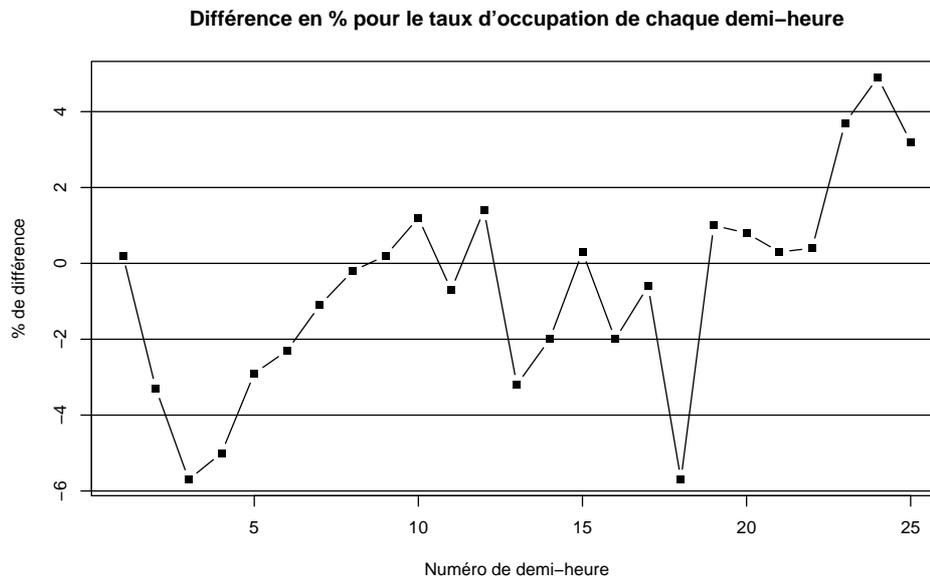


Figure 3.5 – Différence entre le pourcentage empirique et simulé du taux d'occupation des agents pour chaque demi-heure.

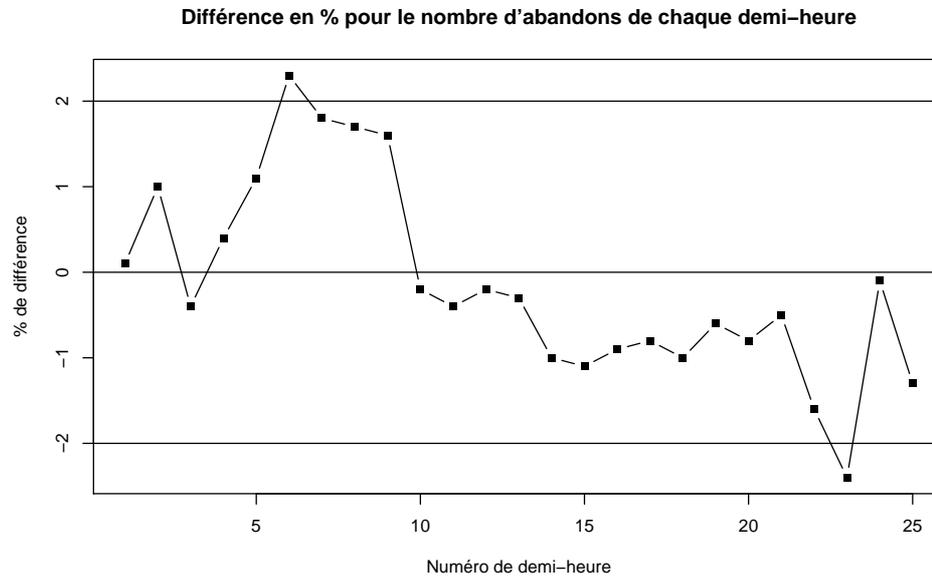


Figure 3.6 – Différence relative entre la valeur empirique et simulée du nombre moyen d’abandons pour chaque demi-heure.

trap”. Nous voulons voir si les différences absolues entre les mesures simulées et empiriques sont assez significatives pour rejeter l’hypothèse que notre modèle est réaliste. Nous allons dénoter par d_i la différence absolue entre la mesure simulée Υ_i de la i -ième demi-heure et la mesure empirique. Si nous connaissions la distribution de la différence absolue des mesures sous l’hypothèse d’avoir un modèle réaliste alors on pourrait évaluer le réalisme de la valeur d_i obtenue. Cette distribution n’est pas connue, mais nous pouvons l’approximer à l’aide du “bootstrap” : nous allons calculer B différences absolues entre Υ_i et la mesure $\Upsilon_{i,k}^*$ provenant de la k -ième simulation du modèle sur un nombre de jours identique à ce que l’on retrouve dans les données. Soit la différence absolue $d_{i,k}^* = |\Upsilon_i - \Upsilon_{i,k}^*|$ de la k -ième simulation, nous obtenons une approximation de la p -valeur :

$$p \approx \frac{\text{Card}\{1 \leq k \leq B : d_{i,k}^* \geq d_i\}}{B}, \quad (3.2)$$

où Card définit la cardinalité d’un ensemble. Cette p -valeur approxime la probabilité d’observer un écart plus grand que d_i en supposant que le modèle est réaliste.

Nous avons appliqué ce test avec $B = 1000$ simulations. Les approximations des p -valeurs des cinq mesures d’intérêt sont présentées dans le tableau 3.3. La valeur ϵ signifie que nous n’avons pas observé de $d_{i,k}^*$ supérieur à d_i , ainsi nous pouvons conclure que la vraie

p -valeur doit être très petite. Les résultats de ce test statistique montrent plusieurs valeurs qui ne sont pas assez petites pour rejeter statistiquement le réalisme de notre modèle. Cependant, il y a des mesures pour certaines demi-heures qui ne semblent pas réalistes. Ceci nous montre que notre modèle est assez réaliste en moyenne sur une journée mais qu'il présente quelques défauts pour la représentation du comportement sur chaque demi-heure. Il est tout de même satisfaisant d'avoir obtenu ce réalisme lorsque l'on pense à la quantité d'informations incomplètes ou manquantes lors de la modélisation.

La simulation de 10000 jours nous demande 72 secondes (moyenne sur 5 exécutions) sur un processeur AMD Athlon de 1750 Mhz. Bien que nous croyons qu'il n'est pas nécessaire d'ajouter de nouveaux détails, nous pensons qu'il est possible d'améliorer le réalisme du modèle. C'est surtout le cas pour les aspects dont nous avons très peu ou aucune information. Ce gain pourrait être significatif si on se fie à un article de Tanir et Booth [51], qui montrent le gain obtenu lorsqu'on modélise le processus d'arrivée avec des données plus raffinées. Cet article nous laisse croire que nous pourrions améliorer notre modèle de la même manière. Le seul problème est que les informations plus raffinées ne sont pas disponibles.

La simulation de ce modèle nous permet de connaître des informations supplémentaires sur le centre d'appels. Nous pouvons, par exemple, en apprendre sur l'attente d'un client. Ceci est intéressant pour Bell, car les données donnent très peu de détails sur la distribution de l'attente d'un client dans le centre d'appels. À titre d'exemples, les simulations nous disent que :

1. La moyenne du temps d'attente d'un client avant que son service débute est de $12,8 \pm 0,3$ secondes (intervalle de confiance à 95%).
2. La moyenne du temps d'attente d'un client en sachant qu'il doit attendre est de $71,8 \pm 0,7$ secondes (intervalle de confiance à 95%).
3. Nous pouvons aussi voir si le centre d'appels respecte d'autres critères de qualité de service. Par exemple, si la règle 80-20 était remplacée par la règle 90-30, i.e. d'avoir au moins 90% des clients qui se font servir en moins de 30 secondes d'attente, alors la simulation nous permet de connaître si cette nouvelle règle est respectée par le centre d'appels. Dans notre cas, on voit que la politique de gestion actuelle ne respecte pas cette règle, car nous obtenons un pourcentage de $87,9 \pm 0,1\%$ des appels entrants se faisant servir après moins de 30 secondes d'attente.
4. Beaucoup d'autres informations peuvent être obtenues via la simulation. Pour les

Demi-heure	Vol. entrant	Vol. sortant	Occupation	Qualité service	% abandons
8h00-8h30	0,012	—	0,862	0,167	0,916
8h30-9h00	€	—	0,004	0,313	0,043
9h00-9h30	€	—	€	0,001	0,378
9h30-10h00	€	—	€	0,846	0,417
10h00-10h30	€	—	0,004	0,394	0,067
10h30-11h00	€	—	0,022	0,001	€
11h00-11h30	0,002	—	0,284	0,059	0,010
11h30-12h00	0,067	—	0,849	0,012	0,011
12h00-12h30	0,341	—	0,857	0,096	0,017
12h30-13h00	0,562	—	0,263	0,355	0,702
13h00-13h30	0,407	—	0,461	0,009	0,074
13h30-14h00	0,499	—	0,168	0,696	0,670
14h00-14h30	0,044	—	€	0,909	0,218
14h30-15h00	0,814	0,140	€	€	€
15h00-15h30	0,706	0,371	0,554	€	€
15h30-16h00	0,872	€	€	€	€
16h00-16h30	0,520	0,006	0,152	€	€
16h30-17h00	0,357	€	€	€	€
17h00-17h30	0,004	0,014	0,055	0,137	0,007
17h30-18h00	0,006	0,508	0,019	0,588	0,004
18h00-18h30	€	0,689	0,434	0,444	0,052
18h30-19h00	0,022	0,021	0,262	0,170	€
19h00-19h30	0,009	€	€	€	€
19h30-20h00	0,114	€	€	€	0,839
20h00-20h30	0,003	€	€	€	0,003

Tableau 3.3 – Approximation des p -valeurs de cinq mesures de performance par demi-heure du modèle.

calculer, il suffit de construire les statistiques s'y rattachant.

Il est important de noter que nous réussissons uniquement à imiter le comportement du centre d'appels étudié. Nous n'avons pas construit un modèle parfaitement réaliste, mais nous croyons qu'il est fort acceptable compte tenu de notre manque d'informations sur plusieurs aspects importants du centre d'appels.

Dans cette sous-section, nous avons déjà proposé d'étudier notre modèle sur chaque jour de la semaine afin d'éliminer certaines mauvaises mesures sur la qualité de service des appels entrants. Un nombre d'agents disponibles qui s'ajuste mieux à l'intensité du trafic d'une journée spécifique semble permettre une meilleure estimation des mesures. Cette division n'est pas sans raison, car nous avons vu dans le chapitre 2 que le taux d'arrivée moyen et la disponibilité des agents varient d'un jour de semaine à l'autre. Nous allons maintenant observer les mesures de notre modèle en modifiant uniquement les paramètres du processus d'arrivée et la disponibilité des agents. Pour le processus d'arrivée d'une journée, nous allons observer les arrivées pour cette journée et utiliser les valeurs des paramètres qui maximisent la fonction de vraisemblance. Et pour la disponibilité des agents, nous allons utiliser les valeurs calculées dans les tableaux 2.16 et 2.17.

Les résultats de la simulation de notre modèle avec les différents paramètres sont présentés dans les tableaux 3.4 à 3.6. Dans la majorité des cas, on ne remarque pas vraiment de différences importantes entre les mesures journalières simulées et empiriques. Il y a tout de même la journée du lundi qui montre des écarts significatifs pour les mesures de qualité de service des appels entrants, i.e. le pourcentage de clients devant attendre plus de 20 secondes et le nombre d'abandons. Nous avons déjà supposé que ce phénomène pourrait être dû à un processus d'arrivée avec une trop forte corrélation entre le nombre d'arrivées dans chaque demi-heure d'une même journée. Est-il possible que ceci explique le comportement du lundi? Dans la sous-section 2.1.1, l'analyse de notre processus d'arrivée nous avait permis de souligner la possibilité que celui-ci introduise inutilement une forte corrélation afin d'ajuster un trafic d'arrivées plus variable. De plus, cette même sous-section montre que le nombre d'arrivées par demi-heure pour le lundi est plus variable par rapport aux autres jours de la semaine. Ainsi, la forte variabilité du nombre d'arrivées pour le lundi pourrait expliquer les écarts observés pour la qualité de service des appels entrants. Pour le lundi, on observe que les corrélations empiriques sont généralement (73 % des cas) plus faibles que les corrélations théoriques du processus définies par la formule (2.22). On observe même que les

	lundi		mardi	
	Simulées	Empiriques	Simulées	Empiriques
Appels entrants reçus	1414,9 ± 5,2	1428,5 ± 82,7	1230,4 ± 4,1	1228,1 ± 67,1
Abandons	63,0 ± 1,4	39,2 ± 8,3	26,9 ± 0,7	28,1 ± 4,8
Appels composés	1512,3 ± 6,8	1540,8 ± 180,8	1954,2 ± 5,0	1783,7 ± 218,1
Appels sortants servis	464,9 ± 2,0	477,4 ± 56,1	601,6 ± 1,5	565,3 ± 69,6
“Mismatches”	33,6 ± 0,2	33,5 ± 4,7	44,4 ± 0,2	38,5 ± 6,2
Occupation des agents	74,6 ± 0,1 %	74,5 ± 2,5 %	71,0 ± 0,1 %	71,7 ± 2,8 %
Qualité de service	78,7 ± 0,1 %	84,1 ± 3,3 %	88,3 ± 0,1 %	87,9 ± 2,4 %

Tableau 3.4 – Tableau comparatif entre les mesures journalières obtenues par la simulation d’un modèle mathématique spécifique à chaque jour de semaine (lundi et mardi) et celles observées dans la réalité. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

différences entre les corrélations théoriques et empiriques sont supérieures à .2 dans plus 20 % des cas. Cette trop forte corrélation jumelée avec le trafic élevé pour le lundi pourraient expliquer les écarts obtenues. Cette petite analyse montre qu’il est important d’avoir les meilleures propriétés possibles pour le processus d’arrivée. Peut-être qu’un processus avec des corrélations indépendantes de la variabilité du nombre d’arrivées pourrait éliminer ce problème. Cette question devrait être abordée dans la suite du projet.

Malgré les quelques différences obtenues, nous pouvons dire que notre modèle semble assez robuste. En modifiant uniquement les paramètres du modèle, nous avons pu l’appliquer aux différents jours de la semaine. Nous avons calculé les p -valeurs de toutes les mesures d’intérêt spécifiques à chaque jour et dans la très grande majorité, les p -valeurs obtenues ne sont pas assez faibles pour rejeter l’hypothèse que notre modèle est réaliste.

D’après ces résultats, nous croyons que notre modèle, dénoté par modèle R (réaliste), est prêt à être utilisé dans l’expérimentation par simulation. En d’autres mots, nous croyons pouvoir analyser le comportement du centre d’appels étudié sous différents scénarios (modification du processus de composition, modification du nombre d’agents, ...) en simulant ceux-ci sur le modèle R. Ces expériences seront décrites dans la prochaine section.

	mercredi		jeudi	
	Simulées	Empiriques	Simulées	Empiriques
Appels entrants reçus	1134,5 ± 3,6	1133,6 ± 56,4	1152,3 ± 3,7	1149,5 ± 58,8
Abandons	21,0 ± 0,5	28,1 ± 4,9	27,7 ± 0,6	33,8 ± 8,7
Appels composés	1622,1 ± 4,5	1679,5 ± 183,1	1653,9 ± 4,6	1765,3 ± 204,4
Appels sortants servis	493,2 ± 1,3	503,8 ± 49,1	505,6 ± 1,4	522,4 ± 57,6
“Mismatches”	36,5 ± 0,2	36,4 ± 4,6	35,8 ± 0,2	38,4 ± 4,9
Occupation des agents	69,2 ± 0,1 %	71,7 ± 2,6 %	69,9 ± 0,1 %	71,8 ± 2,1 %
Qualité de service	90,0 ± 0,1 %	86,8 ± 2,7 %	88,5 ± 0,1 %	85,9 ± 2,6 %

Tableau 3.5 – Tableau comparatif entre les mesures journalières obtenues par la simulation d’un modèle mathématique spécifique à chaque jour de semaine (mercredi et jeudi) et celles observées dans la réalité. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

	vendredi	
	Simulées	Empiriques
Appels entrants reçus	1110,8 ± 3,4	1102,8 ± 52,4
Abandons	27,8 ± 0,6	30,8 ± 6,1
Appels sortants composés	1477,1 ± 4,6	1535,6 ± 191,7
Appels sortants servis	454,3 ± 1,3	447,9 ± 54,0
“Mismatches”	32,3 ± 0,2	35,2 ± 4,1
Occupation des agents	70,7 ± 0,1 %	71,7 ± 2,1 %
Qualité de service	87,0 ± 0,1 %	86,0 ± 2,6 %

Tableau 3.6 – Tableau comparatif entre les mesures journalières obtenues par la simulation d’un modèle mathématique pour le vendredi et celles observées dans la réalité. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

3.1.3 Difficultés rencontrées

Nous allons maintenant faire un bref survol des difficultés que nous avons rencontrées lors de la construction de la librairie de simulation et du développement du modèle. Au début du projet, nous avons construit une première librairie de simulation de centre d'appels. L'idée était de fournir à l'utilisateur un exécutable avec un choix prédéfini d'options. Ces options étaient les différentes lois modélisant les aspects du centre et un choix d'algorithmes pour le système de composition d'appels sortants. Nous pensions que la structure de la librairie permettait une grande flexibilité sur le modèle à simuler. Nous avons appris à nos dépens que ce n'était pas vraiment le cas puisque les mises à jour du modèle étaient assez laborieuses. Le principe d'options facilitait l'utilisation, mais il alourdissait le développement et l'expérimentation de nouveaux modèles. Et comme nous avons principalement besoin d'un outil permettant l'expérimentation, nous avons décidé de construire une nouvelle librairie avec une structure modulaire plus souple. Cette restructuration nous a permis d'avoir une librairie plus flexible, mais elle est devenue moins évidente à utiliser pour une personne avec peu de connaissances en simulation et en programmation. On pourrait faciliter son utilisation en construisant un module pour l'utilisateur qui chapeauterait la librairie. Une autre alternative serait de refaire la librairie en Java. Ceci permettrait d'obtenir une librairie avec une programmation-objet plus claire. Cette traduction et reconception en Java pourrait faire partie d'un projet de maîtrise futur.

Ensuite, nous avons passé beaucoup de temps à tester des heuristiques permettant de bien modéliser le système de composition d'appels sortants. Le pire est que nous avons trop souvent expérimenté ces heuristiques avec des hypothèses erronées sur le centre d'appels. Durant une partie du projet, nous pensions que tous les agents à partir de 14h00 servaient des appels entrants et sortants. Nous avons vu dans le chapitre 2 que nous avons plutôt deux groupes d'agents : un servant uniquement des appels entrants et un autre servant des appels entrants et sortants. Les données disponibles à cette époque ne nous permettaient pas de déduire ce détail. Ainsi, nous avons fait plusieurs tests avec un modèle faux. Nos résultats montraient une trop bonne performance au niveau du nombre d'appels sortants servis, car il y avait trop d'agents pouvant servir ce type d'appels [21]. Les expérimentations ainsi que les analyses faites avec ce modèle n'ont été d'aucune utilité. Nous avons dû tout recommencer avec un modèle corrigé. Ceci montre l'importance d'avoir une bonne interprétation des données que l'on recueille sur le centre d'appels. Suite à cette mauvaise expérience, nous

avons demandé à Bell de nous fournir des données brutes ou non filtrées.

Finalement, les autres difficultés sont surtout survenues lors du calibrage du modèle. Il était particulièrement difficile de fixer les lois pour les aspects dont nous disposions d'aucune ou de très peu d'informations. Malgré ces obstacles, nous avons pu développer un modèle qui semble tout de même assez réaliste.

3.2 Expérimentations

Les modèles décrits dans ce chapitre vont nous permettre d'obtenir les mesures de performance du centre d'appels sous différents scénarios. La modification de l'algorithme de composition d'appels sortants et de la planification du nombre d'agents sont des exemples de scénarios intéressants à analyser.

La simulation est un outil populaire pour faire ce type d'analyse. Il est assez facile de modifier le scénario dans le centre d'appel et d'obtenir ses mesures de performance en simulant de nouveau le modèle. Ensuite, il est possible d'évaluer l'impact des scénarios en comparant les mesures de performance obtenues. Nous avons appliqué la méthode des variables aléatoires communes afin de réduire la variance dans la comparaison des mesures [37]. Cette méthode consiste à soumettre les modèles à des données d'entrée identiques, i.e. les mêmes instants d'arrivée d'appels, durées de service, etc. Notre librairie ainsi que la librairie SSC facilitent l'application de cette méthode. Nous avons une génération indépendante des différents aspects, car nous avons défini un générateur aléatoire pour chacun d'entre eux. De plus, la génération d'une variable d'un jour à l'autre est indépendante, car nous initialisons, à chaque début de journée, notre générateur à un nouveau bloc spécifique (voir module Rand de SSC [40]). Ainsi, les différents scénarios n'influencent pas la génération des variables aléatoires et sont soumis aux mêmes données d'entrées.

3.2.1 Expérimentation #1 : le respect d'une qualité de service à court terme vs à long terme

La première expérimentation servira à comparer différents degrés de restrictions d'une contrainte de qualité de service que l'on peut appliquer dans le centre d'appels. Par exemple, on peut exiger qu'au moins un certain pourcentage de clients entrants doivent attendre

moins de b secondes. On peut forcer le centre d'appels à atteindre cette exigence à court ou à long terme. Nous allons démontrer dans cette expérimentation à quel point on obtient des performances différentes lorsque l'on tente de respecter l'exigence sur une période de temps plus ou moins longue. La motivation de comparer les restrictions sur différentes périodes de temps vient du fait que le système actuel semble utiliser des restrictions à court terme. Cette affirmation s'appuie sur le fait que Mosaix se base, entre autres, sur le pourcentage de clients entrants ayant attendu moins de 20 secondes au cours des 10 dernières minutes. De plus, il y a aussi le gestionnaire d'accès qui modifie l'agressivité de Mosaix dès que la qualité de service se détériore à court terme. Nous allons tenter de montrer que la gestion actuelle de Mosaix et des employés fixant ses paramètres nuit à la performance du centre d'appels.

Le modèle R fait respecter la règle 75-20 sur les 10 dernières minutes et un pourcentage de "mismatches" inférieur à 8% durant les 15 dernières minutes. Il serait intéressant de voir l'effet d'avoir une restriction sur une plus longue période, car il n'y a rien qui indique que nous devions satisfaire ces restrictions sur cette courte période. En fait, il se peut que la règle 80-20 doive être respectée seulement par trimestre. La vraie restriction n'a pas pu être connue, mais elle semble être sur une longue période. Ceci nous a donné l'idée d'utiliser une nouvelle politique de composition n'observant pas de qualité de service en temps réel. Celle-ci respecterait tout juste la qualité de service sur horizon infini. Nous allons en proposer une qui respecte la règle 80-20 sur horizon infini et un nombre quotidien d'abandons et de "mismatches" acceptable. Nous pensons que cette politique moins restrictive permettra d'améliorer l'efficacité du centre d'appels. Dans notre expérimentation, nous allons supposer que le centre est obligé de respecter la règle 80-20 à la fin de chaque trimestre. La nouvelle politique proposée ne peut garantir que cette contrainte sera respectée. Afin de garantir la qualité de service avec probabilité proche de 1 à la fin du trimestre, nous proposons de modifier l'agressivité du compositeur pour les derniers jours du trimestre. Nous sommes confiants que ceci nous permettra d'améliorer la performance du centre d'appels.

Nous allons maintenant décrire la méthode utilisée pour développer notre nouveau processus de composition. Dans un premier temps, nous avons testé un système qui compose, à chaque fin de service, un nombre d'appels égal au double du nombre d'agents libres dans le groupe B. La simulation de notre modèle avec ce processus de composition montre une qualité de service acceptable sur 10000 jours : on obtient $81,9 \pm 0,1\%$ des clients qui attendent

moins de 20 secondes. Cette modification nous permet de servir, en moyenne par jour, 68 (+13%) clients sortants de plus à comparer du modèle R. Bien que nous respections la règle 80-20, nous croyons que les nombres quotidiens d'abandons ($42,4 \pm 0,9$) et de "mismatches" ($84,0 \pm 0,2$) obtenus sont trop élevés. Nous n'avons pas de restriction définie sur ces aspects, mais nous voulons, tout de même, obtenir des valeurs plus raisonnables.

	Mesures avec nouveau processus de composition
Appels sortants servis par jour	$561,8 \pm 1,4$ appels (+49,9)
Abandons par jour	$39,7 \pm 0,9$ abandons (+7,9)
Appels sortants composés par jour	$1810,5 \pm 4,6$ appels (+144,5)
Nombre de "mismatches" par jour	$41,8 \pm 0,2$ "mismatches" (-4,2)
Occupation des agents (%)	$72,9 \pm 0,1$ % (+1,4%)
Qualité de service	$83,0 \pm 0,1$ % (-3,3%)

Tableau 3.7 – Tableau des mesures journalières obtenues par la simulation du modèle R avec un nouveau processus de composition d'appels : compose 2 fois le nombre d'agents libres du groupe B dans 45% des cas et 1 fois ce nombre d'agents libres dans 55% des cas. La qualité de service est définie par le pourcentage d'appels entrants ayant attendu moins de 20 secondes.

Le prochain processus de composition proposé compose, à chaque fin de service, un nombre d'appels égal à 2 fois le nombre d'agents libres du groupe B dans 45% des cas et à 1 fois ce même nombre d'agents libres dans 55% des cas. Les mesures de performance obtenues avec ce processus de composition sont présentées dans le tableau 3.7. Les valeurs entre parenthèses montrent la différence entre les mesures simulées avec ce nouveau processus de composition et celles simulées avec le processus actuel. Nous avons simulé 9984 jours (soit 156 trimestres de 64 jours de semaine). La qualité de service, le nombre d'abandons et le nombre de "mismatches" avec cette nouvelle politique sont données au tableau 3.7.

Sur les 156 trimestres, il y en a seulement 6 qui n'ont pas respecté la règle 80-20. Nous pourrions proposer une correction de l'agressivité du compositeur afin de respecter presque toujours la règle 80-20 à la fin d'un trimestre. Nous avons étudié cette possibilité à l'aide de quelques expérimentations et nous nous sommes rendus compte qu'il fallait diminuer trop sévèrement l'agressivité du compositeur pour améliorer de peu la qualité de service sur un trimestre. Ceci peut s'expliquer par le fait qu'il n'y a pas d'influence sur la qualité de

service pour la période de la journée n'effectuant pas d'appels sortants. Ainsi, il nous faut réduire énormément le nombre d'appels sortants servis pour atteindre la règle 80-20 à la fin du trimestre. Une autre alternative serait de planifier des agents supplémentaires pour les derniers jours du trimestre tant que la règle 80-20 n'est pas respectée. Il faudrait voir si cet ajout d'agents vaut l'augmentation du nombre d'appels sortants servis. Nous avons expérimenté cette méthode en ajoutant 4 agents supplémentaires dans le groupe I pour les 10 derniers jours du trimestre tant que la règle 80-20 n'est pas respectée. Cette nouvelle gestion est avantageuse, car elle nous permet de respecter la règle 80-20 sur 155 des 156 trimestres. Le respect de la contrainte a nécessité 50 jours avec des agents supplémentaires. Les résultats obtenus sont présentés dans le tableau 3.8. On peut résumer qu'en moyenne par trimestre, on planifie 0,32 jours avec 4 agents supplémentaires pour servir environ 3232 appels sortants supplémentaires et perdre environ 499 appels entrants de plus. En supposant que les clients des appels perdus vont sûrement appeler de nouveau, les mesures de performance nous montrent que cette gestion est plus performante que celle actuellement utilisée.

	Mesures avec nouveau processus de composition
Appels sortants servis par jour	562,3 ± 1,4 appels (+50,5)
Abandons par jour	39,6 ± 0,9 abandons (+7,8)
Appels sortants composés par jour	1812,7 ± 4,6 appels (+146,7)
Nombre de "mismatches" par jour	42,0 ± 0,2 "mismatches" (-4,4)
Occupation des agents (%)	73,0 ± 0,1 % (+1,5%)
Qualité de service	83,0 ± 0,1 % (-3,3%)

Tableau 3.8 – Tableau des mesures journalières obtenues par la simulation du modèle R avec un nouveau processus de composition d'appels : compose 2 fois le nombre d'agents libres du groupe B dans 45% des cas et 1 fois ce nombre d'agents libres dans 55% des cas + ajout de 4 agents si la règle 80-20 n'est pas respectée dans les 10 derniers jours d'un trimestre. La qualité de service est définie par le pourcentage d'appels entrants ayant attendu moins de 20 secondes.

Il se peut que la méthode d'ajouter des agents à la fin d'un trimestre ne soit pas vraiment réalisable dans le contexte du centre d'appels étudié. Dans ce cas, il suffit de proposer une méthode mieux adaptée. Nous pourrions utiliser, par exemple, un système de composition d'appels qui atteint une qualité de service supérieure à 83 % (tableau 3.7). Le plus

important à retenir de cet exercice est que nous avons montré qu'on ne doit pas imposer des restrictions sur une courte durée si elles ne sont pas nécessaires. Dans nos expérimentations, une gestion moins restrictive permet d'augmenter le nombre d'appels servis d'environ 10% avec sensiblement les mêmes ressources. Nos résultats montrent que Bell aurait à gagner en modifiant leur processus de composition, i.e. en changeant l'algorithme de Mosaix et en ne modifiant pas ses paramètres par l'entremise des gestionnaires d'accès. Une augmentation des appels servis représente des profits supplémentaires. De plus, il faut rappeler que le centre d'appels étudié est assez petit. La proportion d'appels servis supplémentaires dans un plus gros centre d'appels donnerait des gains encore plus importants.

3.2.2 Expérimentation #2 : variation du nombre d'agents disponibles

Dans cette sous-section, nous allons étudier la sensibilité de notre modèle face à des changements dans la disponibilité des agents. En d'autres mots, nous voulons voir comment les mesures de performance sont affectées par un changement des ressources disponibles. Dans le chapitre 2, nous avons vu que le pourcentage d'appels entrants qui attendent moins de 20 secondes est plus élevé (86,3%) que la restriction de la règle 80-20. Au point de vue de cette contrainte, nous avons une sur-planification des agents. Il serait intéressant de voir l'effet d'une diminution des agents, car on pourrait vouloir éliminer cette sur-planification d'agents.

	Mesures avec 96% des agents
Appels sortants servis par jour	467,9 ± 1,7 appels (-44,0)
Abandons par jour	39,8 ± 1,0 abandons (+8,0)
Appels sortants composés par jour	1519,3 ± 5,3 appels (-146,7)
Nombre de "mismatches" par jour	34,3 ± 0,2 "mismatches" (-3,3)
Occupation des agents (%)	72,6 ± 0,1 % (+1,1%)
Qualité de service	83,4 ± 0,1 % (-2,9%)

Tableau 3.9 – Tableau des mesures journalières obtenues par la simulation du modèle R avec 96% des agents disponibles. La qualité de service est définie par le pourcentage d'appels entrants ayant attendu moins de 20 secondes.

Nous avons utilisé le modèle R pour examiner l'effet d'utiliser 4% moins d'agents au cours de la journée. Cette diminution des agents donne une qualité de service simulée égale

à 83,4%. Ce résultat montre que l'on respecte encore la règle 80-20. Les autres mesures simulées avec cette disponibilité d'agents sont présentées dans le tableau 3.9. La valeur entre parenthèses dénote la différence entre la mesure simulée du modèle R avec 96% des agents et celle avec 100% des agents. Ces résultats nous permettent d'évaluer si une diminution de 4% des agents vaut la peine, i.e si la baisse d'appels sortants et la hausse d'abandons sont compensées par une économie de 4% de salaires d'agents. Toutefois, il serait important d'observer le comportement de la qualité de service simulée tout au long de la journée. Les figures 3.7 et 3.8 comparent la qualité de service simulée avec les deux quantités d'agents disponibles. On remarque une grande dégradation de la qualité de service pour les demi-heures 5 à 9, i.e. entre 10h00 et 12h30. Il se produit même que nous avons moins de 70% des clients qui attendent moins de 20 secondes. Ceci montre qu'une faible variation dans le nombre d'agents peut donner une très mauvaise qualité de service pendant une période critique. C'est à Bell de définir si ce comportement est acceptable ou non. Une façon assez simple d'éliminer la mauvaise qualité de service durant cette période serait de diminuer le nombre d'agents seulement après 14h00, car le système de composition force une qualité de service acceptable. Justement, le tableau 2.15 semble indiquer qu'un grand nombre d'agents débutent leur journée à 13h00. Peut-être serait-il souhaitable de les faire commencer un peu plus tôt afin qu'ils améliorent la performance des périodes critiques. Ces agents devront quitter plus tôt, mais le système de composition va se charger de faire respecter la qualité de service. Ce genre de scénario pourrait améliorer la performance du centre d'appels étudié et sera possiblement examiné à l'aide de la simulation dans la suite du projet.

En supposant que les figures 3.7 et 3.8 montrent une qualité de service acceptable, nous pouvons évaluer à quel point il est possible de réduire le nombre d'agents afin de respecter la règle 80-20. Nous respectons la règle en diminuant le nombre d'agents de 7,6%. Nous présentons les mesures de performance simulées dans le tableau 3.10 et les figures 3.9 et 3.10.

Nous pouvons résumer que la plus importante détérioration de la qualité de service se produit durant la période où il n'y a pas de composition d'appels sortants, i.e. entre 8h00 et 14h00 et aussi qu'il est possible de conserver une bonne qualité en réduisant le volume d'appels sortants composés pour la période entre 14h00 et 20h30. L'augmentation du nombre d'agents disponibles présente le comportement inverse.

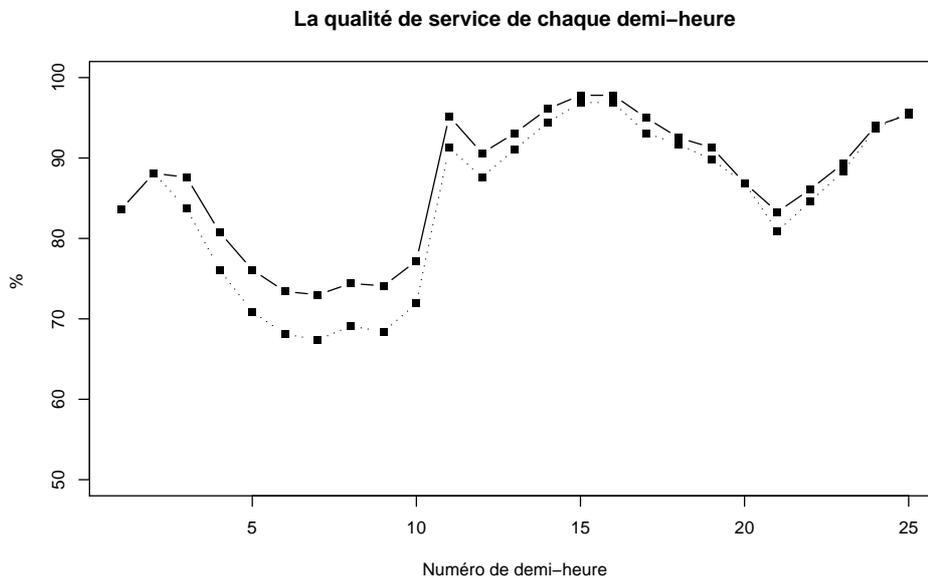


Figure 3.7 – Pourcentage simulé d'appels entrants ayant attendu moins de 20 secondes pour chaque demi-heure avec 96% (ligne pointillée) et 100% (ligne pleine) des agents disponibles.

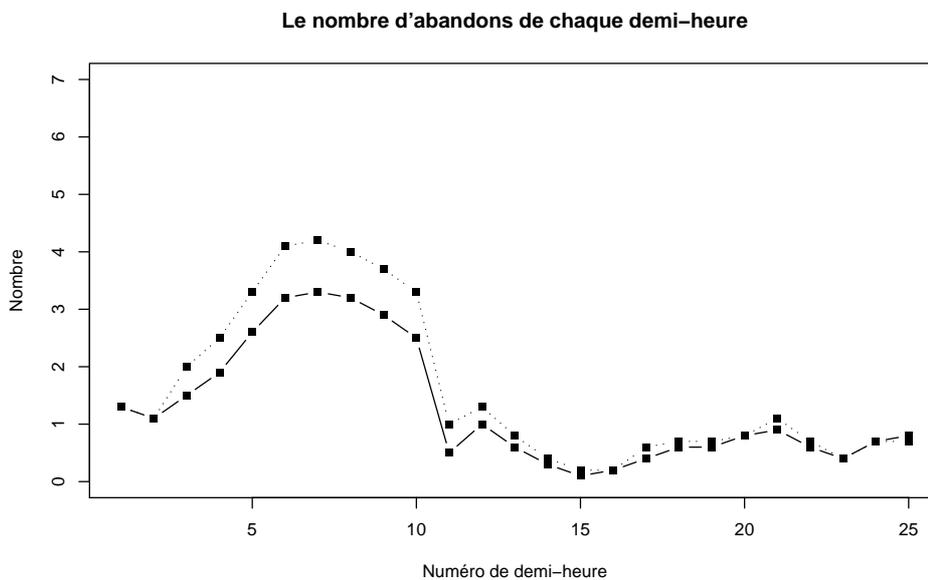


Figure 3.8 – Nombre moyen d'abandons simulé pour chaque demi-heure avec 96% (ligne pointillée) et 100% (ligne pleine) des agents disponibles.

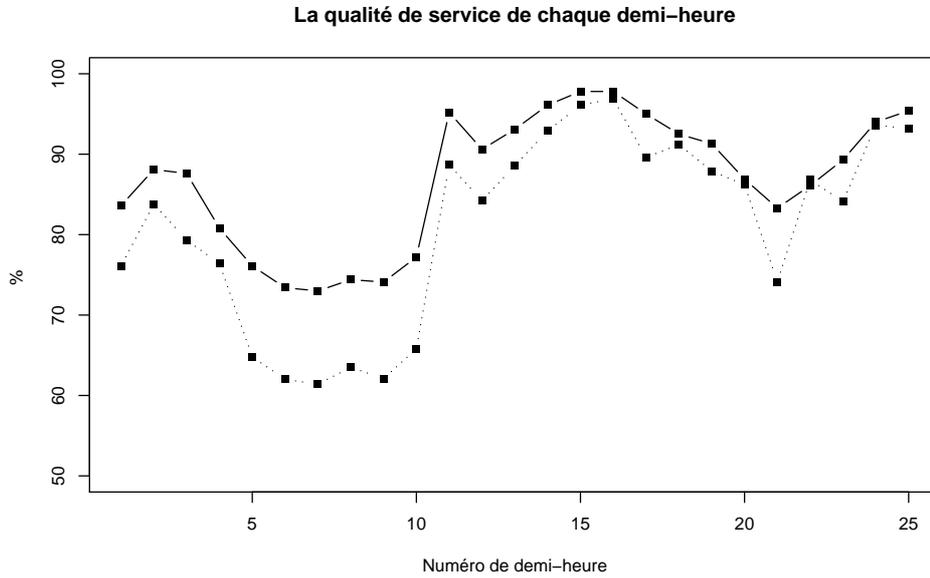


Figure 3.9 – Pourcentage simulé d’appels entrants ayant attendu moins de 20 secondes pour chaque demi-heure avec 92,4% (ligne pointillée) et 100% (ligne pleine) des agents disponibles.

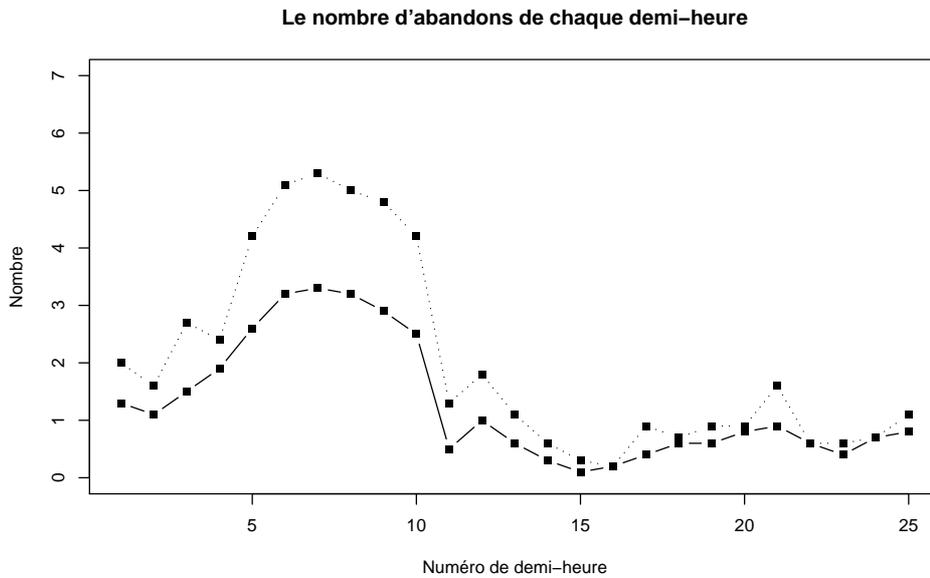


Figure 3.10 – Nombre moyen d’abandons simulé pour chaque demi-heure avec 92,4% (ligne pointillée) et 100% (ligne pleine) des agents disponibles.

	Mesures avec 92,4% des agents
Appels sortants servis par jour	426,6 ± 1,7 appels (-85,3)
Abandons par jour	50,8 ± 1,1 abandons (+19,0)
Appels sortants composés par jour	1391,9 ± 5,7 appels (-274,1)
Nombre de “mismatches” par jour	30,4 ± 0,2 “mismatches” (-7,2)
Occupation des agents (%)	73,1 ± 0,1 % (+1,6%)
Qualité de service	80,1 ± 0,1 % (-6,2%)

Tableau 3.10 – Tableau des mesures journalières obtenues par la simulation du modèle R avec 92,4% des agents disponibles. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

3.2.3 Expérimentation #3 : modifications d’hypothèses dans le modèle

Il peut être tentant de modifier les hypothèses de notre modèle afin de ne pas avoir à utiliser la simulation pour calculer des mesures de performance. C’est le cas lorsque l’on veut utiliser un modèle de CMTC. Ce type de modèle sera étudié au chapitre 4 et nous sommes curieux de connaître le niveau de réalisme de ses hypothèses (voir chapitre 4). Les principales différences, avec notre modèle R, sont des durées de service de loi exponentielle et des arrivées pour chaque demi-heure selon un processus de Poisson stationnaire. La simulation peut être utilisée pour analyser l’effet de ces nouvelles hypothèses. Il suffit de comparer les mesures de performance simulées à l’aide des différents modèles.

Dans cette expérimentation, nous avons construit un autre modèle, dénoté C_1 , sur les bases de notre modèle R. Le modèle C_1 a un processus d’arrivée de Poisson stationnaire par demi-heure où le processus d’arrivée du modèle R avec un W constant égal à 1. Ce processus étudié à la sous-section 2.1.1 n’ajuste pas très bien les données, mais il simplifie l’étude des modèles de CMTC. Les taux d’arrivée de chaque demi-heure sont présentés dans le tableau 2.1. De plus, nous utilisons des durées de service de loi exponentielle. Pour chaque demi-heure, la moyenne de cette loi est la même que celle des lois du modèle R pour les appels entrants et sortants. Le modèle C_1 tient compte des plus importants changements d’hypothèses que nous allons faire dans nos modèles de CMTC du chapitre 4.

Les mesures de performance obtenues par la simulation du modèle C_1 sur 10000 jours sont présentées dans le tableau 3.11. Les valeurs entre parenthèses montrent la différence

	Mesures avec modèle C_1
Appels sortants servis par jour	$525,0 \pm 0,7$ appels (+13,1)
Abandons par jour	$20,0 \pm 0,2$ abandons (-11,8)
Appels sortants composés par jour	$1707,5 \pm 2,5$ appels (+41,5)
Nombre de “mismatches” par jour	$38,4 \pm 0,1$ “mismatches” (+6,6)
Occupation des agents (%)	$72,7 \pm 0,1$ % (+1,2%)
Qualité de service	$89,9 \pm 0,1$ % (+3,6%)

Tableau 3.11 – Tableau des mesures journalières obtenues par la simulation du modèle C_1 . La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

entre les mesures simulées du modèle C_1 et celles du modèle R. Le tableau montre une augmentation importante de la qualité de service simulée par rapport au modèle R. Ceci pourrait être dû à un manque de variabilité dans le nombre d’arrivées par demi-heure. Une augmentation de la variabilité du trafic entraînerait une diminution de la qualité de service. Ces résultats montrent qu’il peut être dangereux d’utiliser les hypothèses du modèle C_1 .

Nous avons refait une simulation avec un autre modèle C_2 dont le processus d’arrivée permet plus de variabilité dans le nombre d’arrivées par demi-heure. Dans ce modèle, nous conservons l’hypothèse d’une loi exponentielle pour les durées de service et nous utilisons le processus d’arrivée Poisson-Gamma de Jongbloed et Koole (sous-section 2.1.1). Nous avons choisi d’expérimenter ce processus, car il sera utilisé sur nos modèles de CMTC du chapitre 4. La simulation de C_2 ne nous donnera pas le comportement exact de ces modèles de CMTC, car ceux-ci calculent leurs mesures sur horizon infini. Malgré cette petite différence, la simulation du modèle C_2 donnera un bon aperçu du réalisme de nos modèles de CMTC avec un processus d’arrivée Poisson-Gamma. Le tableau 3.12 présente les mesures simulées avec ce modèle. Nous remarquons que les mesures simulées sont plus près du modèle R par rapport à celles du modèle C_1 . Les différences observées sont-elles acceptables pour utiliser ces hypothèses? C’est une question de choix et on laisse la réponse aux gestionnaires de Bell.

Dans la dernière simulation, nous allons analyser l’effet de l’hypothèse des durées de service exponentielles. Elle est peut-être responsable des différences observées dans les mesures des modèles C_1 et C_2 . Le tableau 3.13 compare les mesures simulées à l’aide du modèle R

	Mesures avec modèle C_2
Appels sortants servis par jour	$517,0 \pm 0,9$ appels (+5,1)
Abandons par jour	$32,6 \pm 0,3$ abandons (-0,8)
Appels sortants composés par jour	$1677,5 \pm 3,0$ appels (+11,5)
Nombre de “mismatches” par jour	$38,2 \pm 0,1$ “mismatches” (+0,6)
Occupation des agents (%)	$71,9 \pm 0,1$ % (+0,4%)
Qualité de service	$85,5 \pm 0,1$ % (-0,8%)

Tableau 3.12 – Tableau des mesures journalières obtenues par la simulation du modèle C_2 . La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

avec des durées de service exponentielles et celles obtenues avec le modèle R sans modifier les hypothèses sur les durées de service. On remarque qu’il y a peu de différence et il semble que l’effet de cette hypothèse soit assez faible. Il semble toutefois que les différences observées soient plus grandes par rapport à celles observées avec le modèle C_2 . Ceci pourrait être dû au fait que le processus Poisson-Gamma annule mieux l’hypothèse “fausse” d’avoir des durées de service exponentielles.

Nous pouvons conclure que l’utilisation des modèles de CMTTC devrait idéalement être jumelée avec un processus d’arrivée d’appels permettant des intensités de trafic assez variables. Nous allons tenter de tenir compte de cette remarque dans le chapitre 4, mais cet aspect sera approfondi dans la suite du projet avec Bell.

	Mesures avec durées exponentielles
Appels sortants servis par jour	$518,6 \pm 1,6$ appels (+6,7)
Abandons par jour	$32,3 \pm 0,8$ abandons (+0,5)
Appels sortants composés par jour	$1683,7 \pm 5,5$ appels (+17,7)
Nombre de “mismatches” par jour	$38,3 \pm 0,2$ “mismatches” (+0,7)
Occupation des agents (%)	$71,7 \pm 0,1$ % (+0,2%)
Qualité de service	$85,7 \pm 0,1$ % (-0,6%)

Tableau 3.13 – Tableau des mesures journalières obtenues par la simulation du modèle R avec des durées de service exponentielles. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes.

Chapitre 4

Théorie, extensions et expérimentations sur des modèles de CMTc

La simulation n'est pas le seul outil pour calculer des mesures de performance sur des modèles représentant des centres d'appels. Sous des hypothèses différentes de celles du chapitre 3, nous pouvons étudier un centre d'appels à l'aide de modèles de chaînes de Markov en temps continu (CMTc). En fait, la famille des formules d'Erlang pour calculer une distribution du temps d'attente des clients dans une file d'attente se base sur ce type de modèles. Ces formules sont couramment utilisées pour la planification du nombre d'agents nécessaires dans les centres d'appels. Elles ont les avantages d'être facile à appliquer et de fournir rapidement des valeurs exactes sur les mesures de performance d'un centre d'appels. On peut rappeler que la simulation nous permet d'estimer sous forme d'intervalle de confiance ces mêmes mesures. Ensuite, ces valeurs sont souvent utilisées pour fixer la politique de gestion d'un centre. Le centre d'appels étudié en est un bon exemple puisqu'il gère, avec ce type de formules, la planification du nombre nécessaire d'agents afin de respecter une certaine distribution du temps d'attente des clients.

Ces formules ont aussi des inconvénients. Dans [6, 18, 33, 51], on présente une liste des dangers de l'utilisation de ces modèles pour représenter les centres d'appels actuels. La principale raison est que les centres sont devenus trop complexes. Trop souvent, ces modèles

ne s'appliquent pas bien à la réalité et leur manque de réalisme peut provoquer des erreurs de gestion.

Le centre d'appels étudié dans ce mémoire est loin d'être semblable aux modèles des formules d'Erlang. La principale différence vient du fait que nous avons un deuxième trafic associé aux appels sortants. Malheureusement, ces formules n'en tiennent pas compte et ainsi, nous ne pouvons pas appliquer directement ce type de modèles à notre centre d'appels. Toutefois, nous aimons les avantages des modèles CMTC dont la rapidité à obtenir une valeur exacte d'une mesure de performance. C'est pour cette raison que nous allons proposer de nouveaux modèles de CMTC qui s'appliquent mieux à la réalité du centre d'appels étudié. Tout au long de ce chapitre, nous devons garder en tête que le but de ces modèles sera d'aider à la planification du nombre nécessaire d'agents afin de respecter une certaine contrainte sur le temps d'attente des clients. Pour chacun des modèles proposés, nous allons montrer comment calculer la distribution du temps d'attente d'un client. Ce calcul sera utile pour choisir la plus petite quantité de préposés qui permet de satisfaire la contrainte.

Ce chapitre est divisé en cinq sections. Premièrement, nous allons faire un bref rappel des modèles de CMTC sur lesquels les formules d'Erlang se basent. Ensuite, nous allons présenter quelques extensions de ces modèles que l'on retrouve dans la littérature et nous allons voir que celles-ci ne s'appliquent pas vraiment à notre cas. C'est pour cette raison que nous allons proposer nos propres extensions qui s'ajustent mieux à notre centre d'appels. Ces extensions seront, par la suite, utilisées dans des expériences de planification pour ce centre d'appels. Et finalement, nous allons conclure ce sujet en présentant des améliorations possibles à nos modèles.

4.1 File d'attente avec une seule source de clients

Soit $X(t)$ le nombre de clients dans le système au temps $t \geq 0$. Les formules d'Erlang dérivent de la représentation de $\{X(t) \in \{0, 1, 2, \dots\}, t \geq 0\}$ comme un processus de Markov en temps continu ou plus précisément un processus de naissance et de mort [52]. Ce dernier a des probabilités de transition $P_{ij}(t)$ stationnaires; i.e.,

$$P_{ij}(t) = Pr\{X(t+s) = j | X(s) = i\} \text{ pour tout } s \geq 0. \quad (4.1)$$

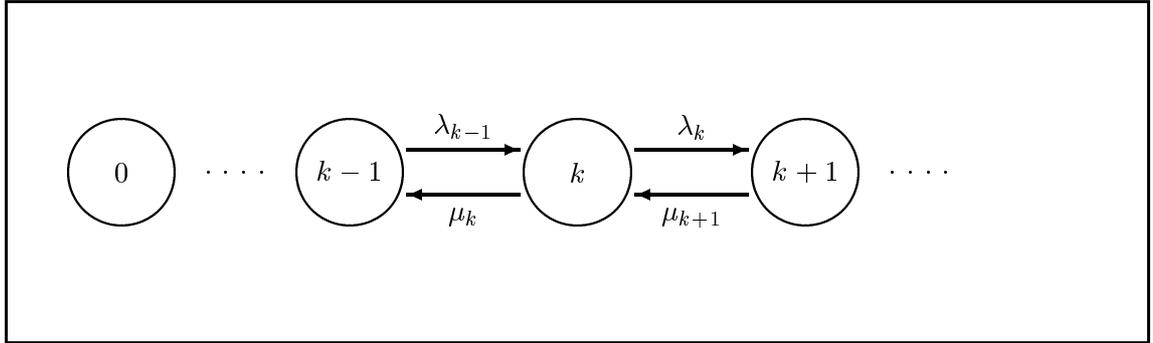


Figure 4.1 – Schéma général d'un processus de naissance et de mort.

Et, nous savons que les $P_{ij}(t)$ satisfont

- (1) $P_{i,i+1}(h) = \lambda_i h + o(h)$ lorsque $h \downarrow 0, i \geq 0$;
- (2) $P_{i,i-1}(h) = \mu_i h + o(h)$ lorsque $h \downarrow 0, i \geq 1$;
- (3) $P_{i,i}(h) = 1 - (\lambda_i + \mu_i)h + o(h)$ lorsque $h \downarrow 0, i \geq 0$;
- (4) $\mu_0 = 0, \lambda_0 > 0, \mu_i, \lambda_i > 0, i = 1, 2, \dots$

où $o(h)$ représente simplement un terme négligeable dans le sens que si nous le divisons par h , alors le résultat tendra vers zéro lorsque h tendra vers zéro [52]. La figure 4.1 donne un aperçu d'un tel processus.

Pour un processus de naissance et de mort avec aucun état absorbant, il est possible de prouver que les probabilités stationnaires ou limites

$$\pi_j = \lim_{t \rightarrow \infty} P_{ij}(t) \quad (4.2)$$

existent et sont indépendantes de l'état initial i [52]. Nous pouvons voir π_j comme étant la fraction du temps que la chaîne passe dans l'état j à long terme. Soient

$$\theta_0 = 1 \text{ et } \theta_j = \frac{\lambda_0 \lambda_1 \cdots \lambda_{j-1}}{\mu_1 \mu_2 \cdots \mu_j} \text{ pour } j \geq 1.$$

Dans [52], on obtient :

$$\pi_0 = \left[\sum_{j=0}^{\infty} \theta_j \right]^{-1} \quad (4.3)$$

$$\pi_j = \theta_j \pi_0, j \geq 1, \quad (4.4)$$

si $\sum_{j=0}^{\infty} \theta_j < \infty$ et on obtient $\pi_j = 0$ pour tout i si $\sum_{j=0}^{\infty} \theta_j = \infty$. Ces équations serviront au développement de la formule d'Erlang-C [19] qui nous permet de calculer une distribution sur le temps d'attente des clients.

Dans la formule d'Erlang-C, on fait l'hypothèse que le centre d'appels est une file $M/M/n$. Voici les principales hypothèses :

- Il y a n serveurs disponibles.
- Les durées de service des clients sont indépendantes et suivent une loi exponentielle de moyenne $1/\mu$.
- Les arrivées des clients suivent un processus de Poisson stationnaire de taux λ .
- Les clients, dans la file d'attente de capacité infinie, n'abandonnent jamais.
- Il n'y a pas de processus de composition pour des appels sortants, i.e., il n'y a aucun trafic sortant.

La file $M/M/n$ est un cas particulier d'un processus de naissance et de mort où $\lambda_i = \lambda$ pour tout i , $\mu_0 = 0$, et $\mu_i = \min(i, n)\mu$ pour $i \geq 1$.

Soit W la durée d'attente d'un client dans la file et $\rho = \lambda/(n\mu)$ l'intensité du trafic, voici la formule d'Erlang-C pour calculer la probabilité qu'un client attende plus que t unités de temps :

$$P\{W > t\} = \begin{cases} P\{W > 0\}e^{-(1-\rho)n\mu t} & , \quad \rho < 1 \\ 1 & , \quad \rho \geq 1, \end{cases} \quad (4.5)$$

où

$$P\{W > 0\} = \frac{(n\rho)^n}{n!(1-\rho)} \left[\sum_{j=0}^{n-1} \frac{(n\rho)^j}{j!} + \frac{(n\rho)^n}{n!(1-\rho)} \right]^{-1} \quad (4.6)$$

est la probabilité qu'un client doive attendre avant de se faire servir. Les détails pour obtenir cette formule sont présentés dans [19].

Cette formule est présentement utilisée pour calculer le nombre d'agents requis au centre d'appels étudié afin de respecter la règle 80-20. On suppose que le centre peut être modélisé par une file $M/M/n$ spécifique à chaque demi-heure. De cette façon, Bell calcule le nombre minimal d'agents, pour chaque demi-heure, leur permettant de respecter la règle 80-20 sur horizon infini. Ensuite, la planification tient compte du trafic sortant en ajustant, si nécessaire, le nombre d'agents à planifier à l'aide d'un facteur de correction. Cette méthode est loin d'être idéale, car on utilise un modèle erroné, pour ensuite le corriger à l'aide d'un facteur de correction. Il est sûrement possible de faire mieux. L'utilisation d'un tel facteur de correction peut engendrer une mauvaise performance du centre d'appels : soit en planifiant trop ou pas assez d'agents.

4.2 Extensions connues dans la littérature

Il existe, dans la littérature, beaucoup d'extensions de cette file $M/M/n$. On peut, par exemple, introduire des abandons de clients dans la file d'attente, un processus d'arrivée non-stationnaire, un deuxième processus d'arrivée, etc. Parmi celles-ci, nous nous sommes principalement attardés aux modèles introduisant un deuxième processus d'arrivée puisque l'on étudie un centre d'appels en mode mixte.

Les articles [10, 11, 12, 13] présentent des modèles qui s'appliquent à un centre d'appels avec deux trafics d'appels. Ces quatre articles présentent des modèles qui sont assez similaires à ce qu'on retrouve dans le centre d'appels étudié.

Dans [10, 11], les auteurs proposent une méthode pour optimiser les règles de la composition d'appels sortants. Leur modèle utilise les mêmes hypothèses que dans une file $M/M/n$, i.e., arrivées du trafic #1 selon un processus de Poisson stationnaire, durées de service i.i.d. de loi exponentielle pour les deux types d'appels, etc. Ils ne font qu'ajouter un processus de composition d'appels sortants. À chaque fois qu'il y a moins de \hat{n} serveurs occupés, le centre rejoint immédiatement un client par un appel sortant. Ceci implique qu'il y a toujours au moins \hat{n} clients dans le centre. De plus, les auteurs supposent qu'il y a une infinité de clients sortants à rejoindre. Le but de l'article est de trouver le seuil \hat{n} qui optimise une certaine fonction objective sous la contrainte de respecter la règle 80-20 pour les appels du trafic #1.

Les deux autres articles [12, 13] se basent essentiellement sur les mêmes hypothèses à l'exception qu'on ajoute le principe du client impatient. En fait, ils supposent que le client i du trafic #1 est prêt à attendre une durée maximale, dénotée U_i , dans la file d'attente. Ces durées U_i sont des variables aléatoires i.i.d. définies par une fonction de répartition générale. Les auteurs se basent sur ce modèle pour présenter différentes manières de calculer les probabilités stationnaires, la probabilité qu'un appel entrant ait à attendre, la probabilité qu'un appel entrant soit perdu à cause de son impatience et des distributions sur le temps d'attente d'un client d'appel entrant. L'idée d'ajouter la notion d'impatience d'un client est intéressante, car elle permet d'améliorer le réalisme du modèle. Ceci permet d'éliminer la sur-planification connue de la formule d'Erlang-C : on suppose qu'il n'y a pas d'abandon dans la file.

Malgré les similitudes de ces modèles avec le centre d’appels étudié, nous n’allons pas appliquer les résultats de ces articles à notre projet. Il y a plusieurs raisons qui expliquent ce choix. Tout d’abord, le processus de composition des appels est trop loin de la réalité. Ces modèles ne tiennent pas compte que la composition d’un appel sortant peut échouer. Cet aspect n’est pas négligeable puisque plus de 50 % des compositions échouent. Ensuite, nous avons décrit que le système de composition peut composer plusieurs appels en parallèle. Ce détail qui entraîne la surcharge du centre, et du même coup les “mismatches”, devrait idéalement être modélisé. La notion de “mismatch” n’existe malheureusement pas dans les modèles décrits dans [10, 11, 12, 13]. Dans ces articles, les auteurs proposent l’utilisation d’un processus de composition simple qui ne représente pas bien notre centre d’appels (voir [7]). Ainsi, nous croyons que la modélisation d’un système de composition plus réaliste serait essentielle à l’étude de notre centre d’appels. Ensuite, il y a aussi le fait que nous avons 2 groupes d’agents. Les modèles présents dans la littérature ne tiennent pas compte de cette subtilité en supposant que tous les agents ont les mêmes tâches. Ces raisons nous ont obligé à proposer de nouveaux modèles qui s’appliquent mieux à notre problème.

4.3 Nouvelles extensions

Cette section est consacrée à la description de nos modèles de chaînes de Markov en temps continu tentant de représenter le centre d’appels étudié. Ceux-ci sont des extensions de modèles que l’on retrouve dans la littérature. Le premier modèle ajoute la possibilité d’avoir une composition d’un appel sortant pouvant échouer. Ensuite, le deuxième modèle, basé sur le premier, ajoute la composition d’appels sortants en parallèle et la notion de “mismatch”. Et finalement, le troisième introduit la modélisation de deux groupes d’agents ainsi qu’une différence entre la loi des durées de service d’appels entrants et celle pour les durées d’appels sortants. Il est certain que le troisième modèle, en ayant le plus de détails, est le plus réaliste, mais nous devons évaluer si l’ajout de ces détails vaut le coût de calcul additionnel. Nous allons tenter de conclure sur le meilleur modèle à utiliser.

Il est important de noter que nos trois modèles évoluent dans un environnement stationnaire pour le processus d’arrivée, le nombre de serveurs, les durées de service, etc. La non-stationnarité sera étudiée dans la section 4.4.

4.3.1 Modèle #1

Le premier modèle est le plus simple et il étend quelque peu le modèle utilisé pour trouver la formule d'Erlang-C (4.5). La principale différence est que l'on ajoute un processus de composition pour les appels sortants et un processus d'abandons. Ce modèle conservera le comportement d'un processus de naissance et de mort, car nous allons utiliser un système de composition d'appels sortants assez simple, des taux de service identiques pour les appels entrants et sortants et des abandons après des durées d'attente exponentielles. Ce modèle fait quelques hypothèses différentes de celles pour la file $M/M/n$. Voici les principales différences :

1. Les durées de service pour les appels entrants et sortants sont i.i.d. de loi exponentielle avec une moyenne $1/\mu$. Soit $1/\mu_e$ la durée de service moyenne des appels entrants et $1/\mu_s$ celle pour les appels sortants. Nous définissons

$$1/\mu = b_e/\mu_e + b_s/\mu_s, \quad (4.7)$$

où b_e est la proportion d'appels entrants servis et b_s est la proportion d'appels sortants servis.

2. La file d'attente a une capacité finie de c places.
3. La probabilité qu'un client n'abandonne pas lorsqu'il ne peut être servi dès son arrivée est dénoté par γ . Et dans le cas où le client n'abandonne pas, on suppose que sa durée maximale d'attente acceptée est exponentielle de taux η .
4. La probabilité de rejoindre un client à l'aide d'un appel sortant est dénotée par κ .
5. Les durées requises pour rejoindre un client à l'aide d'un appel sortant sont exponentielles (i.i.d) de taux ν .
6. Le processus de composition des appels sortants est le suivant : il tente de rejoindre les clients un à la fois. Il compose un appel s'il y a au plus \hat{n} agents occupés et s'il n'y a aucun appel sortant en attente de réponse. Le compositeur arrête sa tentative de rejoindre un client dès qu'il y a plus de \hat{n} agents occupés.

Soit $X(t)$ le nombre de clients dans le système au temps $t \geq 0$. Ces nouvelles hypothèses font en sorte que $\{X(t) \in \{0, 1, 2, \dots\}, t \geq 0\}$ est un processus de naissance et de mort avec

les paramètres suivants :

$$\lambda_k = \begin{cases} \lambda + \kappa\nu & , \quad k = 0, 1, \dots, \dot{n} \\ \lambda & , \quad k = \dot{n} + 1, \dots, n - 1 \\ \gamma\lambda & , \quad n \leq k < n + c \\ 0 & , \quad k \geq n + c \end{cases} \quad (4.8)$$

et

$$\mu_k = \begin{cases} k\mu & , \quad k = 0, 1, 2, \dots, n - 1 \\ n\mu + (k - n)\eta & , \quad k \geq n. \end{cases} \quad (4.9)$$

Avec ce modèle, nous allons montrer qu'il nous est possible de trouver une formule analytique permettant de calculer $P\{W > t\}$. Une forme analytique permettra d'obtenir la distribution de W dans un temps de calcul rapide.

Démarche pour trouver la formule $P\{W > t\}$

Étant donné que $\{X(t), t \geq 0\}$ est un processus de naissance et de mort, nous pouvons calculer les probabilités limites π_k à l'aide des équations (4.3) et (4.4). Les paramètres (4.8) et (4.9) nous permettent d'obtenir :

$$\pi_k = \begin{cases} \left[\sum_{j=0}^{\infty} \theta_j \right]^{-1} & , \quad k = 0 \\ \frac{1}{k!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^k \pi_0 & , \quad 1 \leq k \leq \dot{n} \\ \frac{1}{k!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{k - \dot{n} - 1} \pi_0 & , \quad \dot{n} < k < n \\ \frac{1}{n!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{n - \dot{n} - 1} \prod_{l=1}^{k-n} \frac{\gamma\lambda}{n\mu + l\eta} \pi_0 & , \quad n \leq k \leq n + c \\ 0 & , \quad k > n + c, \end{cases} \quad (4.10)$$

où

$$\theta_j = \begin{cases} \frac{1}{j!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^j & , \quad 1 \leq j \leq \dot{n} \\ \frac{1}{j!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{j - \dot{n} - 1} & , \quad \dot{n} < j < n \\ \frac{1}{n!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{n - \dot{n} - 1} \prod_{l=1}^{j-n} \frac{\gamma\lambda}{n\mu + l\eta} & , \quad n \leq j \leq n + c \\ 0 & , \quad j > n + c. \end{cases}$$

On obtient π_0 par :

$$\begin{aligned}\pi_0 &= \left[\sum_{j=0}^{\infty} \theta_j \right]^{-1} \\ &= \left[\sum_{j=0}^{\dot{n}} \frac{1}{j!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^j + \sum_{j=\dot{n}+1}^n \frac{1}{j!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{j-\dot{n}-1} + \right. \\ &\quad \left. \frac{1}{n!} \left(\frac{\lambda + \kappa\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{n-\dot{n}-1} \sum_{j=n}^{n+c} \prod_{l=1}^{j-n} \frac{\gamma\lambda}{n\mu + l\eta} \right]^{-1}.\end{aligned}$$

Une fois π_0 connu, les équations (4.10) nous permettent de calculer les probabilités π_k pour $1 \leq k \leq n+c$. Ces π_k sont ensuite utilisés pour calculer la probabilité $P\{W > t\}$. Soit V le nombre de clients dans le système, $P\{V = j+n\}$ la probabilité qu'un client à son arrivée observe $j+n$ clients dans le système et $P\{W > t|V = j+n\}$ la probabilité que le temps d'attente d'un client venant d'arriver et observant j clients en attente soit supérieur à t . Selon la théorie des probabilités conditionnelles, on peut écrire :

$$P\{W > t\} = P\{W > 0\}P\{W > t|W > 0\} \quad (4.11)$$

$$= \sum_{j=0}^c P\{W > t|V = j+n\}P\{V = j+n\}. \quad (4.12)$$

Plusieurs de ces probabilités peuvent être représentées en fonction des probabilités stationnaires π_k . Ceci est possible à cause de la propriété PASTA [57]. Cette propriété nous permet de dire que :

Propriété 4.3.1 PASTA.

Pour tout i , la fraction des arrivées (processus de Poisson) trouvant la chaîne dans l'état i est égale à π_i , la fraction du temps où la chaîne est dans l'état i .

Par cette propriété, on obtient que :

$$\begin{aligned}P\{W > 0\} &= \sum_{j=n}^{n+c} \pi_j \\ &= \frac{1}{n!} \left(\frac{\lambda + \gamma\nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{n-\dot{n}-1} \pi_0 \sum_{j=n}^{n+c} \prod_{l=1}^{j-n} \frac{\gamma\lambda}{n\mu + l\eta}, \quad (4.13)\end{aligned}$$

$P\{V = j+n\} = \pi_{j+n}$ et on a par [46] :

$$P\{W > t|V = j+n\} = e^{-\eta t} \frac{\left(\frac{n\mu}{\eta} \right) \left(\frac{n\mu}{\eta} + 1 \right) \cdots \left(\frac{n\mu}{\eta} + j \right)}{j!} \sum_{k=0}^j (-1)^k \binom{j}{k} \frac{e^{-(n\mu+k\eta)t}}{n\mu/\eta + k}. \quad (4.14)$$

À l'aide des équations (4.13) et (4.14), on peut calculer :

$$P\{W > t|W > 0\} = \sum_{j=0}^{\infty} P\{W > t|V = j+n\} \frac{\pi_{j+n}}{P\{W > 0\}} \quad (4.15)$$

et

$$\begin{aligned}
P\{W > t\} &= \sum_{j=0}^c P\{W > t | V = j + n\} \pi_{j+n} \\
&= \frac{\pi_0 e^{-\eta t}}{n!} \left(\frac{\lambda + \kappa \nu}{\mu} \right)^{\dot{n}+1} \left(\frac{\lambda}{\mu} \right)^{n-\dot{n}-1} \sum_{j=0}^c \prod_{l=1}^j \frac{\gamma \lambda}{n\mu + l\eta} \\
&\quad \frac{\left(\frac{n\mu}{\eta} \right) \left(\frac{n\mu}{\eta} + 1 \right) \cdots \left(\frac{n\mu}{\eta} + j \right)}{j!} \sum_{k=0}^j (-1)^k \binom{j}{k} \frac{e^{-(n\mu+k\eta)t}}{n\mu/\eta + k}. \quad (4.16)
\end{aligned}$$

Le grand avantage de cette formule est qu'elle permet de calculer la distribution de W de manière analytique. L'estimation de cette distribution sera plus réaliste qu'avec la formule (4.5) pour un temps de calcul semblable. Nous allons toutefois voir que le réalisme du modèle #1 a des limites. Le modèle représente bien un centre d'appels en mode mixte, mais la simplification du compositeur place ce modèle assez loin de ce que l'on observe dans la réalité.

4.3.2 Modèle #2

Ce nouveau modèle tentera de mieux représenter le processus de composition d'appels sortants que l'on retrouve dans le centre d'appels étudié. Voici les différences au niveau des hypothèses pour ce modèle par rapport au modèle #1 :

1. La durée requise pour rejoindre un client à l'aide d'un appel sortant est comprise dans la durée de service. Ainsi, nous supposons que le service d'un client débute immédiatement après la composition de l'appel.
2. Le processus de composition des appels sortants est le suivant : il tente de rejoindre des clients à chaque fois qu'un agent se libère et qu'il y a au plus \dot{n} agents occupés. Si o est le nombre d'agents occupés à l'instant de la composition alors le nombre d'appels composés en parallèle sera dénoté par $v(o)$. On rappelle aussi que chaque appel composé a une probabilité de connexion κ , de sorte que le nombre d'appels connectés suit une loi binomiale de paramètres $v(o)$ et κ . La probabilité d'avoir x appels sortants connectés lorsqu'on effectue une composition avec o agents occupés est :

$$\phi_o(x) = \binom{v(o)}{x} \kappa^x (1 - \kappa)^{v(o)-x} \quad \text{pour } 0 \leq x \leq v(o).$$

3. Nous avons des “mismatches” lorsque nous rejoignons trop de clients sortants. Si nous sommes dans l'état i et que nous avons rejoint x clients sortants alors nous allons avoir $\max(0, i + x - n)$ “mismatches”. Ces clients sont perdus.

Si on dénote par $X(t)$ le nombre de clients dans le système au temps $t \geq 0$, on obtient que $\{X(t), t \geq 0\}$ est une chaîne de Markov en temps continu. De plus, l'hypothèse de ne pas avoir plus de $n + c$ clients dans le système implique que l'espace d'états pour $X(t)$ est fini. Ceci va permettre de trouver plus facilement les probabilités limites stationnaires π_k . Soit

$$\lim_{h \rightarrow 0^+} \frac{1 - P_{ii}(h)}{h} = q_i, \quad (4.17)$$

le taux de sortie de l'état i et

$$\lim_{h \rightarrow 0^+} \frac{P_{ij}(h)}{h} = q_{ij}, \quad i \neq j, \quad (4.18)$$

le taux de transition de l'état i vers l'état j . Nous savons par [52] que les probabilités limites sont déterminées par les équations :

$$\pi_j q_j = \sum_{i \neq j} \pi_i q_{ij}, \quad j = 0, 1, \dots, n + c \quad (4.19)$$

$$\pi_0 + \pi_1 + \dots + \pi_{n+c} = 1. \quad (4.20)$$

Nous allons utiliser ces π_k , $0 \leq k \leq n + c$ pour calculer la valeur de $P\{W > t\}$.

Démarche pour trouver $P\{W > t\}$

La méthode pour calculer la probabilité est sensiblement la même que pour le modèle #1. Encore une fois, nous pouvons appliquer la propriété PASTA et nous obtenons que :

$$P\{W > 0\} = \sum_{j=n}^{n+c} \pi_j.$$

De plus, les probabilités $P\{W > t | W > 0\}$ et $P\{W > t | V = j + n\}$ sont les mêmes que dans le modèle #1 puisque nous avons le même processus stochastique lorsque nous savons que le client doit attendre. Finalement, il suffit de calculer les π_j pour $j = n, \dots, n + c$, pour obtenir la valeur de $P\{W > t\}$. Ces dernières sont calculées en résolvant le système d'équations (4.19) et (4.20).

Pour bien comprendre les q_i et q_{ji} de ce modèle, nous allons faire un exemple avec $i = 3$ en supposant que $i \leq \dot{n}$. Tout d'abord, nous avons $q_{ji} > 0$ seulement pour $j \leq 4$. Par

exemple, pour $j = 2$, nous avons

$$q_{2,3} = \lambda + 2\mu\phi_1(2).$$

Le taux de transition $q_{2,3}$ couvre deux cas. Dans l'état 2, nous observons l'arrivée d'un appel ou la fin du service d'un client combiné à une composition d'appels sortants rejoignant 2 clients. Nous obtenons aussi :

$$q_3 = \lambda + 3\mu - 3\mu\phi_2(1).$$

Dans ce cas, nous quittons l'état 3 lorsqu'il y a soit l'arrivée d'un appel ou la fin du service d'un client. La dernière valeur soustraite représente le cas où on demeure dans l'état 3 malgré la fin de service d'un client : il y a fin de service d'un client combinée à une composition d'appels sortants rejoignant 1 seul client.

Nous avons utilisé l'ensemble de ces relations pour développer ce système d'équations linéaires :

$$\begin{aligned} (\lambda + i\mu - i\mu\phi_{i-1}(1))\pi_i &= \sum_{\substack{j=1 \\ j \neq i}}^{i+1} j\mu\phi_{j-1}(i-j+1)\pi_j + \lambda\pi_{i-1}, & 0 \leq i \leq \dot{n} \\ (\lambda + i\mu - i\mu\phi_{i-1}(1))\pi_i &= \sum_{\substack{j=1 \\ j \neq i}}^{\dot{n}+1} j\mu\phi_{j-1}(i-j+1)\pi_j + (i+1)\mu\pi_{i+1} + \lambda\pi_{i-1}, & i = \dot{n} + 1 \\ (\lambda + i\mu)\pi_i &= \sum_{j=1}^{\dot{n}+1} j\mu\phi_{j-1}(i-j+1)\pi_j + (i+1)\mu\pi_{i+1} + \lambda\pi_{i-1}, & \dot{n} + 1 < i < n \\ (\gamma\lambda + i\mu)\pi_i &= \sum_{j=1}^{\dot{n}+1} j\mu \sum_{k=i-j+1}^{2(n-j+1)} \phi_{j-1}(k)\pi_j + (n\mu + \eta)\pi_{n+1} + \lambda\pi_{n-1}, & i = n \\ (\gamma\lambda + n\mu + (i-n)\eta)\pi_i &= \gamma\lambda\pi_{i-1} + (n\mu + (i+1-n)\eta)\pi_{i+1}, & n < i < n+c \\ (n\mu + (i-n)\eta)\pi_i &= \gamma\lambda\pi_{i-1}, & i = n+c. \end{aligned}$$

Pour obtenir la valeur des probabilités stationnaires π_j , on doit ajouter la condition que $\sum_{k=0}^{n+c} \pi_k = 1$ et résoudre ce système d'équations linéaires. Une fois que ces probabilités sont calculées, nous pouvons obtenir la probabilité d'attente $P\{W > t\}$ par (4.12) et (4.14). Cette méthode propose un modèle plus "réaliste", mais la résolution du système d'équations linéaires sera plus coûteuse en temps de calcul par rapport à la formule analytique du modèle #1.

4.3.3 Modèle #3

Le troisième et dernier modèle proposé tente d'améliorer la modélisation du groupe d'agents et des durées de service. Nous allons introduire la différence entre les deux groupes d'agents : soit un traitant uniquement les appels entrants et un deuxième servant les deux types d'appels. De plus, nous allons éliminer l'hypothèse que les durées de service des deux types d'appels suivent la même loi. Voici les hypothèses de ce modèle qui diffèrent par rapport au modèle #2 :

1. Les durées de service des appels entrants suivent une loi exponentielle de taux μ_e .
2. Les durées de service des appels sortants suivent une loi exponentielle de taux μ_s .
3. Nous modélisons deux groupes d'agents. Il y a n_1 agents dans le groupe I qui sert uniquement des appels entrants et il y en a n_2 dans le groupe B qui sert les deux types d'appels. On définit le nombre total d'agents par $n = n_1 + n_2$.
4. Le processus de composition des appels sortants est le suivant : il tente de rejoindre des clients à chaque fois qu'un agent se libère, qu'il y a au plus \dot{n} agents occupés et qu'au moins 1 agent soit libre dans le groupe B. Si o est le nombre d'agents occupés dans le groupe B à l'instant de composition, alors le nombre d'appels composés en parallèle sera $v(o)$ pour $o \leq \dot{n}$. On suppose aussi que chaque appel composé a une probabilité de connexion κ , de sorte que le nombre d'appels connectés suit la même loi que dans le modèle #2, i.e., une loi binomiale de paramètres $v(o)$ et κ .
5. Encore une fois, nous avons des "mismatches" lorsque nous rejoignons trop de clients sortants. Si nous avons j agents occupés dans le groupe B et que nous avons rejoint $x > 0$ client(s) sortant(s), alors il y a $\max(0, j + x - n_2)$ "mismatches". Ces clients sont perdus.

Soit

$$X(t) = (Y(t), Z(t)),$$

où $Y(t)$ est le nombre total de clients dans le système au temps $t \geq 0$ et $Z(t)$ est le nombre de clients d'appels sortants dans le système au temps $t \geq 0$. Avec $X(t) \in \{(i, j) : 0 \leq i \leq n + c, 0 \leq j \leq n_2\}$ pour tout t , les hypothèses nous permettent de voir que $\{X(t), t \geq 0\}$ est un processus de Markov en temps continu avec un espace d'états finis. Il nous est possible de calculer les probabilités stationnaires $\pi_{(i,j)}$ en résolvant un système d'équations linéaires que nous expliciterons plus bas.

Démarche pour trouver $P\{W > t\}$

Encore une fois, nous pouvons appliquer la propriété PASTA et nous obtenons que :

$$P\{W > 0\} = \sum_{i=n}^{n+c} \sum_{j=0}^{n_2} \pi_{(i,j)} \quad (4.21)$$

et

$$P\{W > t\} = \sum_{i=n}^{n+c} \sum_{j=0}^{n_2} P\{W > t | V = i, V_S = j\} \pi_{(i,j)}, \quad (4.22)$$

où V est le nombre de clients dans le centre d'appels et V_S est le nombre d'appels sortants en service. Cependant, le seul problème est que la probabilité $P\{W > t | V = i, V_S = j\}$ ne s'obtient pas aussi facilement que dans les deux premiers modèles. La fin de service d'un appel entrant versus sortant modifie différemment le taux de fin de service global puisqu'il n'y a que des appels entrants dans la file.

Connaissant déjà comment calculer les $\pi_{(i,j)}$, il ne reste plus qu'à évaluer $P\{W > t | V = i, V_S = j\}$. Au lieu d'obtenir la valeur exacte, nous proposons d'approximer cette probabilité en nous basant sur la formule (4.14) avec un taux de service moyen fixé à $1/\mu = ((i-j)/\mu_e + j/\mu_s)/i$. Nous avons une meilleure approximation de cette probabilité avec ce modèle. Les deux premiers modèles supposaient un taux de service global égal à $b_e/\mu_e + b_s/\mu_s$. L'avantage de cette méthode est qu'elle tient compte du nombre courant de chaque type de clients à servir. Nous avons choisi cette approximation, car la moyenne du taux de service global avant de servir $i-n$ clients doit être assez près de cette valeur.

Avec ce dernier résultat, le calcul de $P\{W > t\}$ est possible par (4.22) en calculant les $\pi_{(i,j)}$ pour $0 \leq i \leq n+c, 0 \leq j \leq n_2$. Celles-ci sont données en résolvant le système d'équations linéaires (4.19) et (4.20), où I dénote la fonction indicatrice :

Pour $0 \leq i \leq \dot{n}$:

$$\begin{aligned} & (\lambda + (i-j)\mu_e + j\mu_s - j\mu_s\phi_{j-1}(1))\pi_{(i,j)} \\ &= \sum_{k=0}^j (i-j+1)\mu_e\phi_{j-k}(k)\pi_{(i-k+1,j-k)} + \sum_{\substack{k=0, k \neq 1 \\ j-k+1 \leq n_2}}^j (j-k+1)\mu_s\phi_{j-k}(k)\pi_{(i-k+1,j-k+1)} \\ & \quad + I_{[(i-1) \geq j]} \lambda \pi_{(i-1,j)}, \quad 0 \leq j \leq \min(i, n_2 - 1), \\ & (\lambda + (i-j)\mu_e + j\mu_s - j\mu_s \sum_{h=1}^{v(j-1)} \phi_{j-1}(h))\pi_{(i,j)} \\ &= \sum_{k=0}^j (i-j+1)\mu_e \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1,j-k)} \end{aligned}$$

$$\begin{aligned}
& + \sum_{\substack{k=0, k \neq 1 \\ j-k+1 \leq n_2}}^j (j-k+1)\mu_{\mathbf{S}} \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k+1)} \\
& + I_{[(i-1) \geq j]} \lambda \pi_{(i-1, j)}, \quad j = n_2, j \leq i.
\end{aligned}$$

Pour $i = \dot{n} + 1$ et $n \neq \dot{n} + 1$:

$$\begin{aligned}
& (\lambda + (i-j)\mu_{\mathbf{e}} + j\mu_{\mathbf{S}} - j\mu_{\mathbf{S}}\phi_{j-1}(1))\pi_{(i, j)} \\
& = \sum_{k=1}^j (i-j+1)\mu_{\mathbf{e}}\phi_{j-k}(k)\pi_{(i-k+1, j-k)} + \sum_{k=2}^j (j-k+1)\mu_{\mathbf{S}}\phi_{j-k}(k)\pi_{(i-k+1, j-k+1)} \\
& \quad + I_{[(i-1) \geq j]} \lambda \pi_{(i-1, j)} + (i-j+1)\mu_{\mathbf{e}}\pi_{(i+1, j)} \\
& \quad + I_{[j+1 \leq n_2]} (j+1)\mu_{\mathbf{S}}\pi_{(i+1, j+1)}, \quad 0 \leq j \leq \min(i, n_2 - 1), \\
& (\lambda + (i-j)\mu_{\mathbf{e}} + j\mu_{\mathbf{S}} - j\mu_{\mathbf{S}} \sum_{h=1}^{v(j-1)} \phi_{j-1}(h))\pi_{(i, j)} \\
& = \sum_{k=1}^j (i-j+1)\mu_{\mathbf{e}} \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k)} \\
& \quad + \sum_{k=2}^j (j-k+1)\mu_{\mathbf{S}} \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k+1)} + I_{[(i-1) \geq j]} \lambda \pi_{(i-1, j)} \\
& \quad + (i-j+1)\mu_{\mathbf{e}}\pi_{(i+1, j)} + I_{[j+1 \leq n_2]} (j+1)\mu_{\mathbf{S}}\pi_{(i+1, j+1)}, \quad j = n_2, j \leq i.
\end{aligned}$$

Pour $\dot{n} + 1 < i < n$ et $n \neq \dot{n} + 1$:

$$\begin{aligned}
& (\lambda + (i-j)\mu_{\mathbf{e}} + j\mu_{\mathbf{S}})\pi_{(i, j)} \\
& = \sum_{k=(i-\dot{n})}^j (i-j+1)\mu_{\mathbf{e}}\phi_{j-k}(k)\pi_{(i-k+1, j-k)} + \sum_{k=(i-\dot{n})}^j (j-k+1)\mu_{\mathbf{S}}\phi_{j-k}(k)\pi_{(i-k+1, j-k+1)} \\
& \quad + I_{[(i-1) \geq j]} \lambda \pi_{(i-1, j)} + (i-j+1)\mu_{\mathbf{e}}\pi_{(i+1, j)} \\
& \quad + I_{[j+1 \leq n_2]} (j+1)\mu_{\mathbf{S}}\pi_{(i+1, j+1)}, \quad 0 \leq j \leq \min(i, n_2 - 1), \\
& (\lambda + (i-j)\mu_{\mathbf{e}} + j\mu_{\mathbf{S}})\pi_{(i, j)} \\
& = \sum_{k=(i-\dot{n})}^j (i-j+1)\mu_{\mathbf{e}} \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k)} \\
& \quad + \sum_{k=(i-\dot{n})}^j (j-k+1)\mu_{\mathbf{S}} \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k+1)} + I_{[(i-1) \geq j]} \lambda \pi_{(i-1, j)} \\
& \quad + (i-j+1)\mu_{\mathbf{e}}\pi_{(i+1, j)} + I_{[j+1 \leq n_2]} (j+1)\mu_{\mathbf{S}}\pi_{(i+1, j+1)}, \quad j = n_2, j \leq i.
\end{aligned}$$

Pour $i = n$ et $n \neq \dot{n} + 1$:

$$(\gamma\lambda + (i-j)\mu_{\mathbf{e}} + j\mu_{\mathbf{S}})\pi_{(i, j)}$$

$$\begin{aligned}
&= \sum_{k=(i-\hat{n})}^j \min(n, (i-j+1))\mu_e \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k)} \\
&\quad + \sum_{k=(i-\hat{n})}^j (j-k+1)\mu_s \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k+1)} + I_{[(i-1)\geq j]}\lambda\pi_{(i-1, j)} \\
&\quad + ((n-j)\mu_e + \eta)\pi_{(i+1, j)} + I_{[j+1\leq n_2]}(j+1)\mu_s\pi_{(i+1, j+1)}, \quad 0 \leq j \leq n_2.
\end{aligned}$$

Pour $i = \hat{n} + 1 = n$:

$$\begin{aligned}
&(\gamma\lambda + (i-j)\mu_e + j\mu_s - j\mu_s) \sum_{h=1}^{v(j-1)} \phi_{j-1}(h)\pi_{(i, j)} \\
&= \sum_{k=(i-\hat{n})}^j \min(n, (i-j+1))\mu_e \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k)} \\
&\quad + \sum_{\substack{k=(i-\hat{n}) \\ k \neq 1}}^j (j-k+1)\mu_s \sum_{h=k}^{v(j-k)} \phi_{j-k}(h)\pi_{(i-k+1, j-k+1)} + I_{[(i-1)\geq j]}\lambda\pi_{(i-1, j)} \\
&\quad + (\min(n, i-j+1)\mu_e + \eta)\pi_{(i+1, j)} + I_{[j+1\leq n_2]}(j+1)\mu_s\pi_{(i+1, j+1)}, \quad 0 \leq j \leq n_2.
\end{aligned}$$

Pour $n < i < n + c$:

$$\begin{aligned}
&(\gamma\lambda + (n-j)\mu_e + (i-n)\eta + j\mu_s)\pi_{(i, j)} \\
&= \gamma\lambda\pi_{(i-1, j)} + ((n-j)\mu_e + (i+1-n)\eta)\pi_{(i+1, j)} \\
&\quad + I_{[j+1\leq n_2]}(j+1)\mu_s\pi_{(i+1, j+1)}, \quad 0 \leq j \leq n_2.
\end{aligned}$$

Et finalement, l'état avec $i = n + c$:

$$((n-j)\mu_e + (i-n)\eta + j\mu_s)\pi_{(i, j)} = \gamma\lambda\pi_{(i-1, j)}, \quad 0 \leq j \leq n_2.$$

Il suffit de résoudre ce système d'équations linéaires avec la condition que

$$\sum_{i=0}^{n+c} \sum_{j=0}^{n_2} \pi_{(i, j)} = 1, \quad (4.23)$$

pour obtenir les probabilités stationnaires. Celles-ci nous permettent d'obtenir la probabilité d'attente $P\{W > t\}$. Ce modèle semble plus "réaliste" que les deux premiers, mais la résolution du système d'équations linéaires sera coûteuse en temps de calcul. Nous devons évaluer si le gain en réalisme l'emporte sur le temps de calcul supplémentaire.

4.4 Expérimentations

4.4.1 Mesures de performance pour ces modèles

Les modèles décrits se basent sur plusieurs simplifications. La plus importante est certainement que l'on suppose un environnement stationnaire pour le processus d'arrivées d'appels entrants, le nombre de serveurs, etc. Les analyses du chapitre 2 montrent bien que cette hypothèse est irréaliste. Il est inutile d'utiliser ces modèles pour représenter le centre d'appels tout au long de la journée. Toutefois, il y a la possibilité de les utiliser demi-heure par demi-heure. Les planificateurs d'horaires de chez Bell font l'hypothèse que le système est stationnaire par demi-heure et ils utilisent la formule (4.5) d'Erlang-C pour calculer le nombre d'agents à planifier pour chaque demi-heure. Le centre d'appels étudié planifie la plus petite quantité d'agents qui permet au modèle $M/M/n$ de respecter la règle 80-20 sur horizon infini. Nous allons utiliser la même technique avec nos modèles, i.e., utiliser le plus petit n respectant la règle 80-20 sur horizon infini. Le modèle #3 n'est pas différent puisque nous allons supposer que le nombre d'agents dans le groupe B est fixé d'avance. Cette hypothèse ne semble pas trop forte puisque ce nombre d'agents dépend de la planification dans les demi-heures précédentes : les agents travaillent en mode mixte pour les 3,5 dernières heures de leur journée. Dans la suite du projet, il serait tout de même intéressant d'étudier une méthode pour optimiser ces deux quantités.

Nous allons calculer quelques mesures de performance sur ces modèles pour connaître à quel point ils représentent bien le centre d'appels étudié. Les mesures intéressantes seront le pourcentage d'occupation des agents, le taux d'appels sortants servis et le taux de "mismatches". Nous allons utiliser ces trois valeurs et leur temps de calcul nécessaire afin de comparer la performance de chaque modèle. Nous allons maintenant décrire comment calculer ces mesures.

Taux d'occupation des agents

On peut calculer le taux d'occupation T_O des agents sur horizon infini pour chaque modèle décrit dans ce chapitre. Les probabilités stationnaires π_i ($\pi_{(i,j)}$ pour modèle #3) nous donnent la fraction du temps où le système est dans l'état i ((i,j) pour modèle #3).

Ainsi, nous pouvons calculer le taux d'occupation des 2 premiers modèles par :

$$T_O = \sum_{i=0}^n \binom{i}{n} \pi_i + \sum_{i=n+1}^{\infty} \pi_i \quad (4.24)$$

$$= \sum_{i=0}^n \binom{i}{n} \pi_i + \left(1 - \sum_{i=0}^n \pi_i\right). \quad (4.25)$$

Et pour le modèle #3 :

$$T_O = \sum_{i=0}^n \sum_{j=0}^{\min(i, n_2)} \binom{i}{n} \pi_{(i,j)} + \sum_{i=n+1}^{\infty} \sum_{j=0}^{n_2} \pi_{(i,j)} \quad (4.26)$$

$$= \sum_{i=0}^n \sum_{j=0}^{\min(i, n_2)} \binom{i}{n} \pi_{(i,j)} + \left(1 - \sum_{i=0}^n \sum_{j=0}^{\min(i, n_2)} \pi_{(i,j)}\right). \quad (4.27)$$

Taux d'appels sortants servis par unité de temps

Le taux d'appels sortants servis par unité de temps sur horizon infini, dénoté par T_S , est calculé à l'aide des probabilités stationnaires de chaque modèle. Le T_S dans l'état i des deux premiers modèles peut être vu comme le taux de service ($\min(i, n)\mu$) des deux types d'appels moins le taux d'arrivée (λ ou $p\lambda$ si $i \geq n$) des appels entrants. Nous pouvons représenter ceci par

$$T_S = \sum_{i=0}^{n-1} \pi_i (i\mu - \lambda) + \sum_{i=n}^{\infty} \pi_i (n\mu - p\lambda - (i-n)\eta). \quad (4.28)$$

Pour le modèle #3, nous avons directement le taux de service des appels sortants : $j\mu_S$ pour l'état (\bullet, j) . Ceci nous donne :

$$T_S = \sum_{i=0}^{\infty} \sum_{j=0}^{\min(i, n_2)} \pi_{(i,j)} j\mu_S. \quad (4.29)$$

Taux de “mismatches” par unité de temps

Le taux de “mismatches” par unité de temps sur horizon infini est dénoté par T_M . L'absence de “mismatch” dans le modèle #1 fait en sorte que nous avons un taux nul, ce qui n'est pas vraiment réaliste. Pour les deux autres modèles, le “mismatch” survient lors de la composition d'appels sortants, i.e., à la fin du service d'un client. Soit la variable aléatoire M_l = nombre de “mismatches” lorsqu'il y a l agents occupés. Si nous sommes dans l'état $i \leq n$ du modèle #2 alors le taux de “mismatches” égale le taux de fin de service ($i\mu$)

multiplié par le nombre espéré de “mismatches” $E[M_i]$, i.e., :

$$T_m = \sum_{i=0}^{\hat{n}+1} \pi_i i \mu E[M_{i-1}], \quad (4.30)$$

où $E[M_i] = \sum_{j=n-i+1}^{v(i)} (j - (n - i)) \phi_i(j)$.

Et pour le modèle #3, nous dénotons la variable aléatoire du nombre de “mismatches” lorsque nous avons l agents occupés dont k dans le groupe B par $M_{l,k}$. Si nous sommes dans l’état (i, j) alors le taux de “mismatches” est égal à :

$$T_m = \sum_{i=0}^{\hat{n}+1} \sum_{j=0}^{\min(i, n_2)} (I_{[j < n_2]} i \mu_e E[M_{i-1, j}] + (i - j) \mu_s E[M_{i-1, j-1}]) \pi_{(i, j)}, \quad (4.31)$$

où I est la fonction indicatrice et

$$E[M_{i, j}] = \sum_{k=\min(n-i, n_2-j)+1}^{v(j)} (k - \min(n - i, n_2 - j)) \phi_j(k). \quad (4.32)$$

Extensions

Nous allons maintenant voir comment il serait possible d’utiliser une CMTC qui n’a pas nécessairement des probabilités de transition stationnaires. Dans le chapitre 2, nous avons montré qu’il semble irréaliste de penser, par exemple, que les arrivées d’une demi-heure suivent un processus de Poisson stationnaire. L’expérience de la sous-section 3.2.3 nous a montré le danger d’utiliser cette stationnarité.

Dans notre extension, nous voulons trouver une manière d’obtenir les mesures de performance de nos modèles de CMTC avec un processus d’arrivée des appels entrants Poisson-Gamma tel que proposé dans [31]. Ce modèle permet une analyse indépendante de chaque demi-heure sauf que nous avons un taux d’arrivée défini par la variable aléatoire Λ suivant une loi gamma. Sa fonction de densité sera définie par $f(\lambda)$. Ces hypothèses nous permettent de voir chaque période de 30 minutes comme un cycle régénératif. Nous allons définir les mesures avec un processus d’arrivée Poisson-Gamma : $P\{W > t\}$ pour la probabilité d’attendre plus de t unités de temps, T'_s pour le taux d’appels sortants servis, T'_m pour le taux de “mismatches” et T'_o pour le taux d’occupation des agents. Avant d’expliquer leur calcul, nous allons redéfinir les mesures des modèles stationnaires avec un taux d’arrivée égal à λ par $P\{W > t | \Lambda = \lambda\}$, $T_o(\lambda)$, $T_s(\lambda)$ et $T_m(\lambda)$. Soit $\bar{\lambda}$ la moyenne de la variable aléatoire Λ , A_t le nombre d’appels entrants ayant attendu plus de t unités de temps durant

une demi-heure et D le nombre d'appels entrants arrivées durant une demi-heure. Nous obtenons :

$$\begin{aligned}
P'\{W > t\} &= \frac{E[A_t]}{E[D]} \\
&= \frac{E[A_t]}{30\bar{\lambda}} \\
&= \frac{1}{30\bar{\lambda}} \int_0^\infty E[A_t|\Lambda = \lambda]f(\lambda)d\lambda \\
&= \frac{1}{30\bar{\lambda}} \int_0^\infty 30\lambda P\{W > t|\Lambda = \lambda\}f(\lambda)d\lambda \\
&= \frac{1}{\bar{\lambda}} \int_0^\infty \lambda P\{W > t|\Lambda = \lambda\}f(\lambda)d\lambda
\end{aligned} \tag{4.33}$$

et

$$T'_s = \int_0^\infty T_s(\lambda)f(\lambda)d\lambda \tag{4.34}$$

$$T'_m = \int_0^\infty T_m(\lambda)f(\lambda)d\lambda \tag{4.35}$$

$$T'_o = \int_0^\infty T_o(\lambda)f(\lambda)d\lambda. \tag{4.36}$$

Les mesures ainsi obtenues tiendront compte d'une quantité d'arrivées plus variable qu'avec un processus de Poisson stationnaire et seront plus réalistes.

4.4.2 Planification des agents

Dans cette sous-section, nous allons proposer une méthode pour planifier le nombre nécessaire d'agents dans notre centre d'appels. Nos modèles de CMTC nous permettent d'obtenir différentes mesures de performance. Ces dernières serviront à proposer une planification demi-heure par demi-heure. Nous allons retenir le nombre minimum d'agents afin de respecter la règle 80-20 sur horizon infini pour chaque demi-heure.

L'utilisation de nos trois modèles nécessite de définir leurs différents paramètres sur chaque demi-heure. Les tableaux 2.2 et 2.3 donnent les paramètres du processus de Poisson stationnaire, tandis que les tableaux 2.5 et 2.6 donnent les paramètres du processus Poisson-Gamma. Le tableau 2.13 donne les probabilités κ de connexion. Nous allons fixer la capacité de la file d'attente à $c = 20$ et la probabilité d'avoir aucune attente sera $\gamma = 0,005$. Cette probabilité semble négligeable mais il est quand même intéressant d'utiliser toute la richesse de nos modèles. Nos modèles pourraient éventuellement être utilisés dans un contexte où

cette probabilité ne serait pas négligeable. Ensuite, le taux de connexion du modèle #1 sera $v = 30$ par minute. Il y a aussi d'autres paramètres qui se basent sur le modèle R du chapitre 3 dont le taux d'abandons : $1/\eta = 0,005/\eta_1 + 0,995/\eta_2$, où η_1 et η_2 sont les deux taux d'abandons possibles (voir module "Incall" de la sous-section 3.1.1). Le tableau 4.1 donne un aperçu des paramètres utilisés pour les durées de service dans les trois modèles. Les taux pour les modèles #1 et #2 sont définis par la règle (4.7). Les taux sont présentés par minute. Finalement, les paramètres reliés à la composition des appels sortants auront la valeur qui permet d'obtenir des mesures de performance semblables à ce qu'on retrouve dans le centre d'appels étudié. Nous allons les définir un peu plus loin.

Nous nous sommes intéressés uniquement aux journées du lundi et du mercredi. Ceci vient du fait que le lundi montre une intensité de trafic particulière et le mercredi représente bien l'intensité rencontrée dans les 4 autres jours de semaine.

Planification des agents pour la période sans appel sortant

Entre 8h00 et 14h00, le centre d'appels ne sert aucun appel sortant. Sans appel sortant, les 3 modèles proposés dans ce chapitre sont équivalents. En d'autres mots, nous avons une file $M/M/n$ avec des abandons dans la file. Ce système est souvent appelé Erlang-A. Pour cette période de la journée, il est clair que le modèle #1 l'emporte sur les deux autres puisqu'on obtient les mêmes mesures pour la fraction du temps de calcul.

Le nombre nécessaire d'agents afin de respecter la règle 80-20 est dénoté par N_{ST} pour le modèle stationnaire et par N_{PG} avec le modèle utilisant le processus d'arrivée Poisson-Gamma. Il est assez simple d'obtenir N_{PG} , car nous pouvons calculer numériquement l'intégrale dans l'équation (4.33). De plus, la formule analytique (4.16) pour $P\{W > t\}$ du modèle #1 aide à obtenir rapidement le calcul de cette intégrale. Nous utilisons l'algorithme d'intégration numérique adaptative que l'on retrouve dans la librairie GSL (GNU Scientific Library) [22].

Le tableau 4.2 montre les planifications obtenues pour les deux jours et les compare avec celles utilisées dans nos simulations (N_{SIM}). Il est important de noter que N_{SIM} n'est pas le nombre d'agents requis pour avoir la règle 80-20 dans le modèle R. Il représente plutôt la planification actuelle du centre d'appels. Les valeurs de N_{SIM} sont l'arrondissement à l'unité des valeurs du tableau 2.16 multipliées par le facteur de correction de l'efficacité

	Modèle #1 et #2	Modèle #3	
Demi-heure	μ	μ_e	μ_s
8h00-8h30	0,101	0,101	0,0
8h30-9h00	0,101	0,101	0,0
9h00-9h30	0,101	0,101	0,0
9h30-10h00	0,101	0,101	0,0
10h00-10h30	0,101	0,101	0,0
10h30-11h00	0,101	0,101	0,0
11h00-11h30	0,101	0,101	0,0
11h30-12h00	0,101	0,101	0,0
12h00-12h30	0,104	0,104	0,0
12h30-13h00	0,104	0,104	0,0
13h00-13h30	0,104	0,104	0,0
13h30-14h00	0,104	0,104	0,0
14h00-14h30	0,111	0,104	0,136
14h30-15h00	0,116	0,105	0,136
15h00-15h30	0,118	0,105	0,136
15h30-16h00	0,118	0,105	0,136
16h00-16h30	0,117	0,105	0,136
16h30-17h00	0,119	0,109	0,136
17h00-17h30	0,119	0,109	0,136
17h30-18h00	0,122	0,109	0,136
18h00-18h30	0,121	0,109	0,136
18h30-19h00	0,127	0,118	0,136
19h00-19h30	0,128	0,118	0,136
19h30-20h00	0,130	0,118	0,136
20h00-20h30	0,130	0,118	0,136

Tableau 4.1 – Paramètres des taux de service utilisés dans les trois modèles de CMTC.

des agents que nous avons décrit dans le modèle R. Ceci revient à dire que nous nous intéressons uniquement à l'efficacité réelle des agents. Ainsi, on doit tenir compte de ce facteur de correction pour savoir le nombre réel d'agents à planifier. Cette remarque est valable pour le reste de la section.

	lundi			mercredi		
Demi-heure	N_{SIM}	N_{ST}	N_{PG}	N_{SIM}	N_{ST}	N_{PG}
8h00-8h30	12	13	15	11	10	12
8h30-9h00	18	18	19	17	15	16
9h00-9h30	22	23	26	21	19	20
9h30-10h00	25	26	28	23	22	23
10h00-10h30	27	28	31	23	22	23
10h30-11h00	26	29	30	24	23	24
11h00-11h30	26	28	30	23	22	24
11h30-12h00	24	28	30	24	23	24
12h00-12h30	22	26	28	22	20	22
12h30-13h00	23	26	28	21	20	21
13h00-13h30	28	27	30	27	22	23
13h30-14h00	25	26	29	26	20	22

Tableau 4.2 – Tableau comparatif entre le nombre (N_{SIM}) d'agents utilisé dans la simulation du modèle R et les nombres nécessaires d'agents pour respecter la règle 80-20 sur horizon infini du modèle #1 avec un processus d'arrivée stationnaire (N_{ST}) et avec un processus d'arrivée Poisson-Gamma (N_{PG}). Nous considérons chaque demi-heure entre 8h00 et 14h00 du lundi et mercredi.

Ce tableau montre que le nombre d'agents N_{ST} est toujours inférieur à N_{PG} . La plus grande variabilité dans le nombre d'arrivées du processus Poisson-Gamma force la planification de plus d'agents. Nous avons vu dans la sous-section 3.2.3 que le processus de Poisson stationnaire par demi-heure donne une trop bonne qualité de service. Ainsi, le nombre d'agents N_{ST} risque d'être trop bas pour satisfaire la qualité de service dans le centre d'appels réel.

Nous croyons que le nombre N_{PG} donnera une meilleure planification par rapport à N_{SIM} . Nous allons tenter de répondre à cette question en simulant le modèle R du chapitre

3 avec N_{PG} agents disponibles. Mais avant tout, il nous faut calculer les besoins en agents des demi-heures après 14h00. Le tableau 4.2 montre aussi une planification d'un grand nombre d'agents pour la première demi-heure. Ceci est dû au calcul sur horizon infini qui ne tient pas compte d'un système vide au début de cette demi-heure. Nous pourrions corriger ce nombre par un certain facteur.

Planification des agents pour la période avec appels sortants

L'utilisation des trois modèles pour la période entre 14h00 et 20h30 est plus complexe. Nous devons fixer certains paramètres pour la composition d'appels sortants. Le seuil maximal \hat{n} d'agents occupés permettant la composition d'appels sortants, le nombre $v(o)$ d'appels composés lorsqu'il y a o agents occupés et le nombre n_2 d'agents dans le groupe B doivent être définis. Nous avons déjà discuté que nous supposons avoir un nombre d'agents dans le groupe B prédéfini : nous allons prendre les valeurs que nous avons utilisées dans la simulation du modèle R. Pour \hat{n} et $v(o)$, nous allons calculer les mesures de performance avec différentes valeurs et retenir celles qui semblent les plus réalistes.

Pour le modèle #1, nous avons fait des expériences nous montrant qu'il n'y avait aucun algorithme de composition permettant d'obtenir des mesures de performance réalistes. Premièrement, il n'y a aucun "mismatch". Ensuite, on obtenait jamais un nombre moyen d'appels servis par demi-heure près de ce qu'on observe en réalité. Et finalement, le modèle montrait toujours un taux d'occupation des agents trop élevé. Ce modèle ne semble pas vraiment utile pour cette période de la journée. Le modèle #1 est assez semblable à ceux que nous avons décrit dans la section 4.2. Ainsi, les résultats obtenus montrent que nous avons bien fait de rejeter l'application de modèles existants dans la littérature.

Le modèle #2 montre lui-aussi certains problèmes au niveau de ses mesures de performance. Son plus grand défaut est qu'il ne différencie pas les deux groupes d'agents. En d'autres mots, nous faisons l'hypothèse que tous les agents sont dans le groupe B. Ceci implique que nous obtenons des mesures assez réalistes pour les demi-heures entre 18h00 et 20h30, car le nombre d'agents dans le groupe I est proportionnellement faible. Cependant, nous obtenons des mauvaises mesures pour les demi-heures entre 14h00 et 18h00. Quelques expérimentations nous ont permis de montrer que nous avons un système trop performant durant cette période : le nombre d'appels sortants servis est trop grand, le nombre de "mis-

matches” est trop faible et le taux d’occupation des agents est trop grand. Ceci influence le trafic d’appels à servir, la qualité de service et ainsi, la planification du nombre nécessaire d’agents pour respecter la règle 80-20. Encore une fois, il semble que ce modèle ne soit pas applicable.

Le modèle #3 règle la plupart des problèmes rencontrés dans les deux premiers modèles. Cependant, ce gain en réalisme est coûteux en temps de calcul. En fait, l’optimisation du nombre nécessaire d’agents pour une demi-heure en particulier peut prendre plus de 2 heures. Ceci ne nous empêche pas d’utiliser cette méthode. Dans le cas pratique, on pourrait utiliser des tables pré-calculées pour chaque demi-heure qui décriraient pour un sous-ensemble de α et β du processus d’arrivée le nombre d’agents à planifier. Sans nécessairement entrer dans les détails, nous voulons simplement montrer que cette méthode de planification est applicable dans la réalité.

Tout d’abord, nous devons fixer les paramètres de composition du modèle #3. Pour le lundi, nous obtenons des mesures assez réalistes pour le nombre de d’appels sortants servis par demi-heure et le nombre de “mismatches” par demi-heure avec $\dot{n} = n - 3$ entre 14h00-16h00, $\dot{n} = n - 4$ entre 16h00-19h00 et $\dot{n} = n - 5$ entre 19h00-20h30, et $v(o) = n_2 - o + 1$. Pour le mercredi, nous utilisons le même $v(o)$, mais on fixe $\dot{n} = n - 3$ entre 14h00-17h00 et $\dot{n} = n - 4$ entre 17h00-20h30. Nous n’avons pas la même agressivité pour le processus de composition du lundi et celui du mercredi, car les taux d’arrivée et le nombre d’agents diffèrent entre ces deux jours. Le tableau 4.3 montre le nombre d’agents disponibles N_{SIM} dans la simulation du modèle R et le nombre d’agents à planifier N_{PG} d’après le modèle #3 avec un processus d’arrivée Poisson-Gamma. La troisième colonne de chaque jour donne le nombre d’agents disponibles dans le groupe B.

Selon le modèle #3, nous pouvons réduire le nombre d’agents disponibles dans la grande majorité des demi-heures. N_{SIM} montre une sur-planification allant jusqu’à 7 agents en trop. Cette sur-planification n’est pas une surprise, car nous avons remarqué une trop bonne qualité de service sur certaines demi-heures.

Nous pouvons conclure que la planification pour cette période de la journée semble préférable avec le modèle #3 : les deux premiers modèles donnent des mesures trop loin de la réalité. Il reste à vérifier l’impact de cette planification d’agents. Est-ce qu’on respecte toujours la qualité de service ? Et quelle est son impact sur le volume d’appels sortants ?

	lundi			mercredi		
Demi-heure	N_{SIM}	N_{PG}	groupe B	N_{SIM}	N_{PG}	groupe B
14h00-14h30	30	30	5	27	25	5
14h30-15h00	34	34	11	31	27	10
15h00-15h30	39	36	16	36	29	14
15h30-16h00	41	39	18	39	32	17
16h00-16h30	37	36	16	37	31	16
16h30-17h00	31	32	14	31	29	14
17h00-17h30	26	25	11	26	23	11
17h30-18h00	26	27	16	24	24	15
18h00-18h30	20	24	16	20	21	16
18h30-19h00	19	20	16	18	18	14
19h00-19h30	19	17	16	16	16	13
19h30-20h00	20	18	17	19	16	15
20h00-20h30	19	17	16	17	15	14

Tableau 4.3 – Tableau comparatif entre le nombre d’agents disponibles dans la simulation du modèle R et le nombre nécessaire d’agents pour respecter la règle 80-20 sur horizon infini sur le modèle #3 avec un processus d’arrivée Poisson-Gamma. Nous avons déjà fixé le nombre d’agents dans le groupe B. Nous considérons chaque demi-heure entre 14h00 et 20h30 du lundi et mercredi.

Simulation avec notre planification

Nous allons maintenant étudier le comportement du centre d’appels sous la planification d’agents proposée dans les tableaux 4.2 et 4.3. Nous allons faire la simulation du modèle R décrit dans le chapitre 3 avec N_{PG} agents disponibles. Pour chacune des journées (lundi et mercredi), nous allons pouvoir calculer des mesures de performance sous cette nouvelle gestion et les comparer avec celles simulées sous la gestion actuelle (N_{SIM}). Les tableaux 4.4 et 4.5 montrent la comparaison sur des mesures journalières simulées.

Pour le lundi, nous avons une amélioration de la qualité de service des appels tant au niveau de la distribution de l’attente qu’au niveau des abandons. Ce gain se fait cependant au détriment du taux d’occupation des agents. Ce dernier est plus faible, car la période 8h00-14h00 montre un nombre moyen d’agents plus élevé avec les N_{PG} qu’avec les N_{SIM} .

	Mesures avec nouvelle planification (lundi)
Appels sortants servis par jour	450,3 ± 2,1 appels (-14,6)
Abandons par jour	31,7 ± 0,9 abandons (-31,3)
Appels sortants composés par jour	1462,1 ± 7,0 appels (-50,2)
Nombre de “mismatches” par jour	31,9 ± 0,2 “mismatches” (-1,7)
Occupation des agents (%)	71,7 ± 0,1 % (-2,9%)
Qualité de service	87,6 ± 0,1 % (+9,2%)

Tableau 4.4 – Tableau des mesures de performance obtenues suite à la simulation du modèle R du lundi avec la planification d’agents obtenue à l’aide des modèles de CMTC. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes. Le nombre entre parenthèses est la différence observée avec les mesures simulées sous la planification actuelle.

Il est intéressant de noter que l’amélioration de la qualité de service ne se fait pratiquement pas au détriment du nombre d’appels sortants servis : on améliore un comportement tout en conservant une même performance.

La nouvelle planification du mercredi permet d’augmenter le taux d’occupation des agents. Ceci est principalement dû à une diminution du nombre nécessaire d’agents. Il est important de noter que cette diminution permet toujours de respecter la règle 80-20. Sauf que cette baisse d’agents entraîne une diminution dans le nombre d’appels sortants servis. Cette diminution d’agents vaut-elle la baisse d’appels sortants servis ? Nous laisserons Bell répondre à cette question. Toutefois, nous pouvons mentionner que nos modèles de CMTC calculent le taux d’appels sortants servis et que ce calcul pourrait être utilisé dans une planification sous contrainte d’avoir un certain taux d’appels sortants servis. Cette nouvelle méthode d’optimisation pourrait faire partie de la suite du projet avec Bell.

Les tableaux 4.4 et 4.5 montrent une qualité de service bien au dessus de la limite du 80 %. Ceci est peut-être dû à nos modèles de CMTC qui forcent une planification trop conservatrice, mais nous pourrions aussi l’expliquer par le fait que nous avons un centre d’appels de petite dimension. L’ajout d’un agent supplémentaire a un plus grand impact sur notre centre d’appels que sur un centre ayant plus de 100 agents. C’est un peu pour cette raison que nous avons de la difficulté à atteindre une qualité de service plus près de 80 %.

	Mesures avec nouvelle planification (mercredi)
Appels sortants servis par jour	421,3 ± 1,6 appels (-71,9)
Abandons par jour	26,3 ± 0,6 abandons (+5,3)
Appels sortants composés par jour	1366,8 ± 5,5 appels (-255,3)
Nombre de “mismatches” par jour	29,3 ± 0,2 “mismatches” (-7,2)
Occupation des agents (%)	71,0 ± 0,1 % (+1,8%)
Qualité de service	87,6 ± 0,1 % (-2,4%)

Tableau 4.5 – Tableau des mesures de performance obtenues suite à la simulation du modèle R du mercredi avec la planification d’agents obtenue à l’aide des modèles de CMTC. La qualité de service est définie par le pourcentage d’appels entrants ayant attendu moins de 20 secondes. Le nombre entre parenthèses est la différence observée avec les mesures simulées sous la planification actuelle.

Notre nouvelle planification montre de bons résultats journaliers, mais il pourrait être intéressant de connaître son comportement sur chaque demi-heure. Les figures 4.2 à 4.5 présentent la qualité de service et le taux d’occupation simulés sur chaque demi-heure. La ligne pointillée représente les mesures simulées sous la planification actuelle et la ligne pleine représente les mesures simulées sous notre nouvelle planification. Les figures sur la qualité de service sont les plus intéressantes, car elles montrent une amélioration tout au long de la journée. Nous réussissons à corriger certaines mauvaises qualités de service obtenues sous la planification actuelle (surtout les demi-heures 4 à 10 du lundi).

Les plus importantes diminutions du nombre nécessaire d’agents se produisent pour les demi-heures avec une excellente qualité de service sous la planification actuelle. Nos diminutions en nombre d’agents permettent d’augmenter l’occupation des agents tout en respectant la qualité de service (voir demi-heures 11 à 19 du mercredi). Les figures 4.4 et 4.5 montrent le gain obtenu, sur chaque demi-heure, pour le taux d’occupation des agents.

La planification proposée à l’aide de nos modèles de CMTC semble assez robuste pour l’utiliser dans le processus de construction d’horaire du centre d’appels étudié. La simulation nous a permis de voir qu’il est possible de faire des gains considérables avec une meilleure planification du nombre nécessaire d’agents. La prochaine étape serait d’utiliser notre méthode de planification dans le processus de construction d’horaire pour chaque agent sous différentes contraintes (quart de travail, pause-café, dîner, formation, etc.).

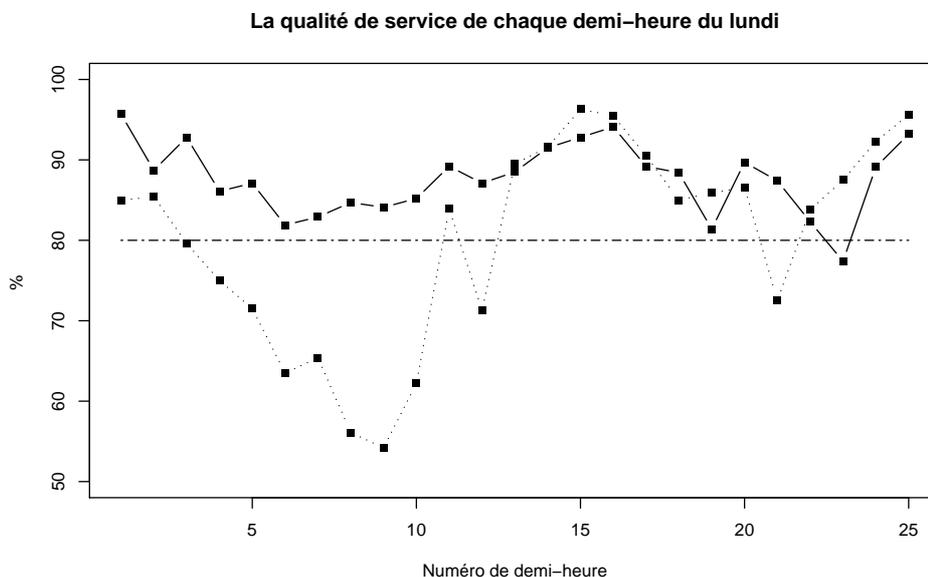


Figure 4.2 – La qualité de service (pourcentage d’appels entrants ayant attendu moins de 20 secondes) simulée pour chaque demi-heure du lundi avec la planification actuelle (ligne pointillée) et la planification obtenue à l’aide des modèles de CMTC (ligne pleine).

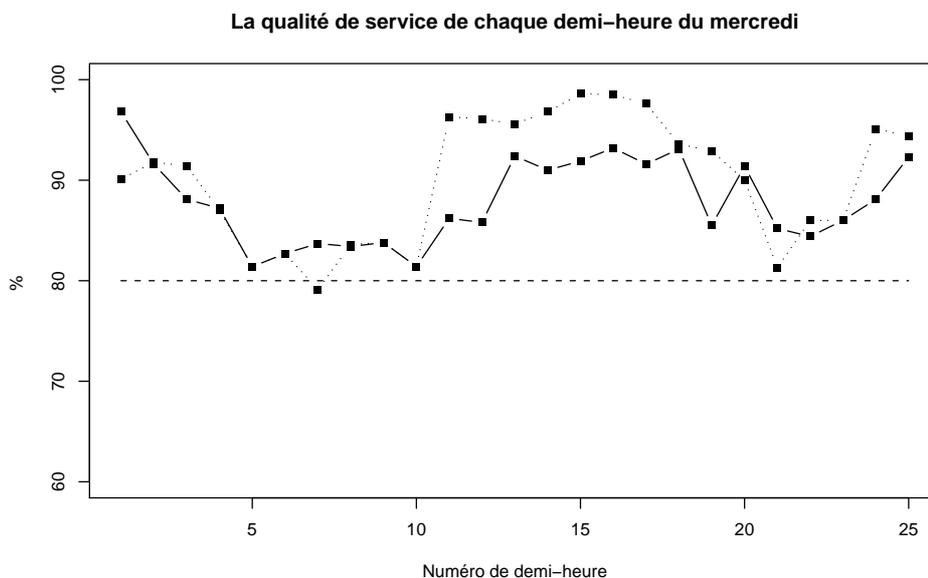


Figure 4.3 – La qualité de service (pourcentage d’appels entrants ayant attendu moins de 20 secondes) simulée pour chaque demi-heure du mercredi avec la planification actuelle (ligne pointillée) et la planification obtenue à l’aide des modèles de CMTC (ligne pleine).

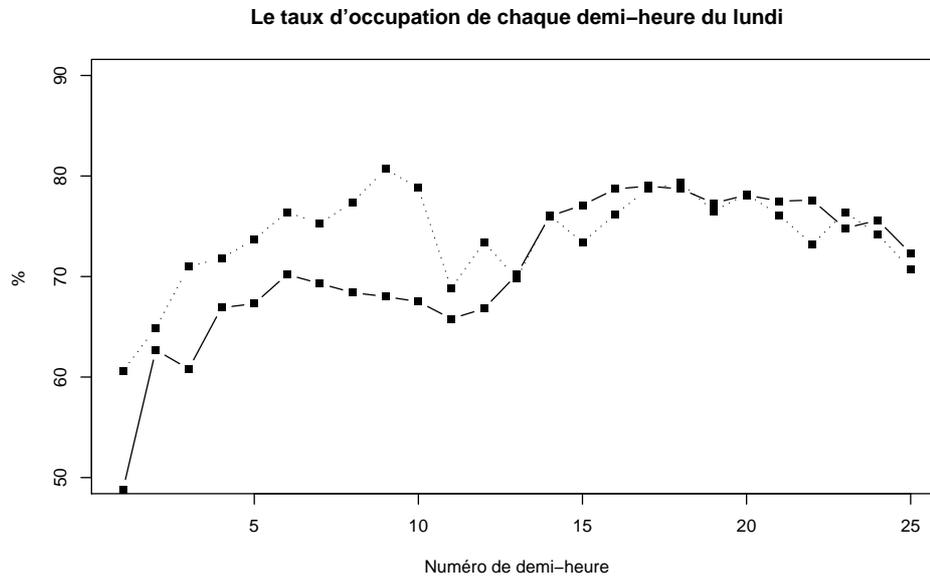


Figure 4.4 – Taux d'occupation des agents simulé pour chaque demi-heure du lundi avec la planification actuelle (ligne pointillée) et la planification obtenue à l'aide des modèles de CMTC (ligne pleine).

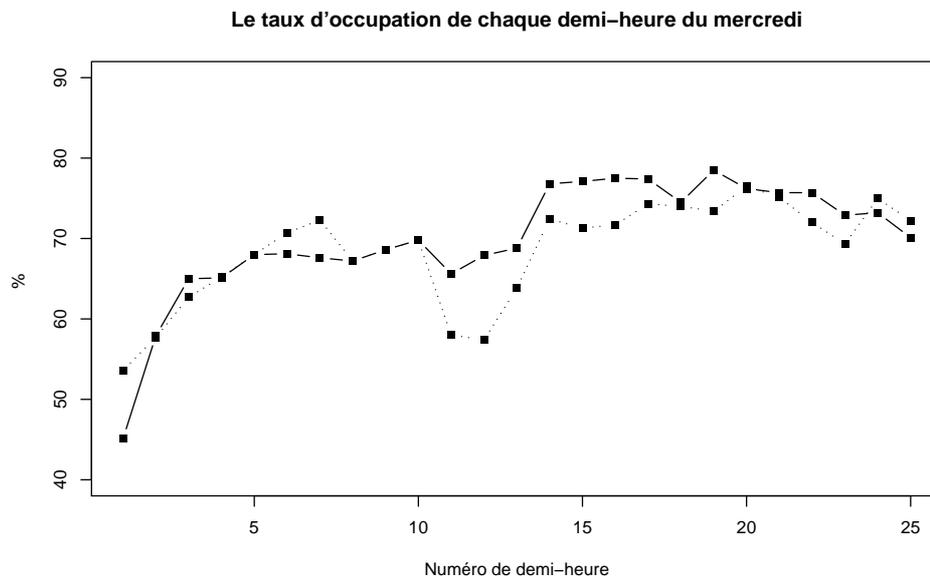


Figure 4.5 – Taux d'occupation des agents simulé pour chaque demi-heure du mercredi avec la planification actuelle (ligne pointillée) et la planification obtenue à l'aide des modèles de CMTC (ligne pleine).

4.5 Extensions possibles

Il existe des extensions possibles à nos modèles de CMTC qui n'ont pas été explorées dans ce mémoire. Les résultats de ce chapitre et l'expérience de simulation de la sous-section 3.2.3 nous montrent l'importance de bien modéliser le processus d'arrivée des appels. Malgré les bons résultats obtenus dans ce mémoire, nous croyons qu'il serait intéressant d'étudier des modèles plus complexes. Dans nos modèles, nous supposons que nous avons des demi-heures indépendantes. Ceci a simplifié nos calculs, mais il est clair que nous avons fait une hypothèse pas vraiment réaliste. Ainsi, nous pourrions nous intéresser à d'autres processus d'arrivée et tenter de les appliquer dans nos modèles. Encore une fois, l'étude de processus plus complexes serait facilitée avec l'obtention de données plus raffinées.

Nos modèles permettent une certaine non stationnarité entre les demi-heures, mais il semble que nous pourrions les améliorer en nous inspirant des modèles $M(t)/M(t)/s$. Il existe des méthodes connues dans la littérature pour le calcul de mesures de performance avec ces modèles. Nous pourrions peut-être les appliquer dans notre cas. Parmi celles-ci, il y a le "Pointwise Stationary Approximation" (PSA) [24, 25, 29, 55] et le "Simple Stationary Approximation" (SSA) [29, 55] aussi connu sous le nom "Average Stationary Approximation" (ASA). Le principe du PSA est de considérer le système à chaque instant t , de faire temporairement l'hypothèse que le modèle est stationnaire pour la valeur de ses paramètres, et de calculer les mesures de performance sur horizon infini ainsi obtenues. On fait cela pour chaque valeur de t , puis on calcule la moyenne par rapport au temps t , sur la période de temps $[0, T)$ qui nous intéresse, de chaque mesure de performance considérée. Cette moyenne par rapport au temps sert à approximer la moyenne correspondante ou l'espérance mathématique pour le système non stationnaire. Pour ce qui est du SSA, il applique simplement un modèle stationnaire avec le taux d'arrivée moyen pour la période. En fait, nous utilisons un peu cette méthode dans ce chapitre. De plus, il serait possible d'explorer une autre méthode d'approximation développée par Jennings, Mandelbaum, Massey et Whitt [29]. Cette méthode, qui se nomme "infinite servers" (IS), a prouvé son utilité lorsque le taux d'arrivée des appels entrants fluctue assez rapidement [29]. Nous croyons qu'il est possible d'évaluer la précision et l'utilité de chacune de ces approximations dans le contexte d'un modèle représentant un centre d'appels en mode mixte.

Ensuite, il serait aussi possible d'améliorer nos modèles au niveau du processus de com-

position des appels sortants. Pour l'instant, nous utilisons un processus assez simple qui est loin de ce qu'on retrouve dans la réalité. Son amélioration pourrait permettre d'obtenir de meilleures mesures et une meilleure gestion. Le problème est que ce processus doit prendre ces décisions avec l'état actuel du système : le rendre plus complexe pourrait demander un énorme espace d'états. Il faudrait étudier dans quelles mesures nous pouvons améliorer le réalisme de la composition des appels sortants en conservant un modèle utilisable.

L'étude de ces extensions semble être la suite logique de nos travaux sur les CMTC.

Chapitre 5

Conclusion

La principale contribution de ce mémoire se trouve dans la modélisation d'un centre d'appels en mode mixte. Premièrement, nous avons tenté d'ajuster des lois de probabilité à différents aspects d'un centre d'appels réel. Cette étape s'est faite sous la forme d'une analyse de données : nous avons fait de l'ajustement de paramètres et testé statistiquement le réalisme de ces ajustements. Dans cette analyse de données, nous nous sommes spécialement attardés au processus d'arrivée des appels entrants. Nous avons étudié quelques processus doublement stochastiques existants de la littérature :

- Processus Poisson-Gamma décrit dans [31]. Celui-ci suppose une indépendance entre les différentes demi-heures. Nous croyons que ce processus est loin d'être idéal, car il ne permet aucune corrélation entre le nombre d'arrivées de demi-heures successives. En plus, il semble que sa large paramétrisation permet un bon ajustement, mais très peu d'informations au niveau de la prévision.
- Processus doublement stochastique décrit dans [56]. Ce processus a un taux d'arrivée stochastique $\Lambda(t)$ de la forme :

$$\Lambda(t) = W\lambda(t), t \geq 0, \tag{5.1}$$

où $\lambda(t) = \lambda_i$ lorsque t est dans la demi-heure i et W est une variable aléatoire positive dépendante de la journée. Nous supposons que W est de loi gamma indépendante des autres processus dans le centre d'appels. Ce processus permet un bon ajustement des données et des nombres d'arrivées corrélés entre les demi-heures et ceci avec une plus petite paramétrisation que le processus Poisson-Gamma.

Nous avons principalement étudié le deuxième processus. Nous avons développé une méthode pour estimer ses paramètres en maximisant sa fonction de vraisemblance. Ensuite, nous avons étudié quelques-unes de ses propriétés. Parmi celles-ci, nous avons vu qu'il est possible de forcer des corrélations trop fortes entre les nombres d'arrivées des demi-heures d'une même journée. Toutefois, ce processus semble assez réaliste pour l'appliquer dans notre cas.

L'analyse de données et l'ajustement des processus d'arrivée nous ont permis de développer des modèles représentant un centre d'appels en mode mixte. Ceux-ci présentent différents niveaux de détails et de réalisme. Dans certains cas, nous avons utilisé la simulation pour obtenir leurs mesures de performance et analyser la sensibilité du centre d'appels lorsqu'on modifie sa gestion. Nous avons aussi construit trois modèles basés sur les CMTC. Ces derniers ont été utilisés pour la planification du nombre nécessaire d'agents afin de respecter la règle 80-20. Ces modèles spécifiques à notre centre d'appels sont des nouvelles extensions de ce que l'on retrouve dans la littérature.

Une autre contribution est la construction d'une librairie permettant la simulation d'un modèle représentant un centre d'appels en mode mixte. Cette librairie, disponible au laboratoire de simulation du département d'informatique et de recherche opérationnelle de l'Université de Montréal, permet d'expérimenter la sensibilité d'un modèle sous différents scénarios. L'ensemble de nos expérimentations propose des améliorations possibles dans la gestion du centre.

En plus de la modélisation, ce mémoire a contribué à fixer les bases d'un projet de 3 ans avec Bell. Il a permis d'obtenir des informations sur le fonctionnement du centre, d'améliorer notre interprétation des données et de comprendre les besoins de Bell dans ce projet. La suite du projet se basera sur les travaux présentés dans ce mémoire.

Bibliographie

- [1] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.
- [2] M. Armory and C. Maglaras. On customer contact centers with a call-back option : Customer decisions, sequencing rules, and system design. Technical report, Graduate School of Business, Columbia University, 2001. Preprint.
- [3] Avaya Communications. Blending : The changing color of contact center productivity, 2001, <http://www1.avaya.com/enterprise/whitepapers/gcc0910.pdf>.
- [4] J. Banks. *Handbook of Simulation*. Wiley, New York, 1998.
- [5] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol. *Discrete-Event System Simulation*. Prentice-Hall, Upper Saddle River, N.J., third edition, 2000.
- [6] V. Bapat and E.B. Pruitte Jr. Using simulation in call centers. In *Proceedings of the 1998 Winter Simulation Conference*, volume 2, pages 1395–1399, 1998.
- [7] N. Belacel, A. Deslauriers, and P. L'Ecuyer. Deuxième rapport d'étape pour le projet LUB : Gestion optimale d'un centre d'appels dans un environnement "blend". GERAD, Université de Montréal, octobre 2001.
- [8] N. Belacel, P. Hansen, and P. L'Ecuyer. Rapport d'étape pour le projet LUB : gestion optimale d'un centre d'appels en environnement "blend", juillet 2001. Rapport fourni à Bell, 28 pages.
- [9] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Volumes I and II*. Athena Scientific, Belmont, Mass., 1995.
- [10] S. Bhulai and G. Koole. A queueing model for call blending in call centers. Technical Report WS-542, Vrije Universiteit, Amsterdam, 2000.
- [11] S. Bhulai and G. Koole. Scheduling time-constrained jobs in the presence of background traffic. *Proceedings of the 39th IEEE Conference on Decision and Control*, 2 :1421–1426, 2000.

- [12] A. Brandt and M. Brandt. On the $M(n)/M(n)/s$ queue with impatient calls. *Performance Evaluation*, 35(1-2) :1–18, 1999.
- [13] A. Brandt and M. Brandt. A two-queue priority system with impatience and its application to a call center. *Methodology and Computing in Applied Probability*, 1 :191–210, 1999.
- [14] P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer-Verlag, New York, second edition, 1987.
- [15] Call Center News Service. Call center statistics, 2001. <http://www.callcenternews.com>.
- [16] G. Casella and R.L. Berger. *Statistical inference*. Duxbury advanced series, second edition, 2002.
- [17] R. Chokshi. Decision support for call center management using simulation. In *Proceedings of the 1999 Winter Simulation Conference*, volume 2, pages 1634–1639, 1999.
- [18] B. Cleveland and J. Mayben. Call center management - on fast forward. Preprint, 1997. Call Center Press.
- [19] R. B. Cooper. *Introduction to queueing theory*. North Holland, second edition, 1981.
- [20] A. Deslauriers, N. Channouf, and P. L'Ecuyer. Rapport d'étape pour le projet LUB : Gestion optimale d'un centre d'appels dans un environnement "blend". Département d'informatique et de recherche opérationnelle, Université de Montréal, Juillet 2002.
- [21] A. Deslauriers, P. Hansen, and P. L'Ecuyer. Troisième rapport d'étape pour le projet LUB : Gestion optimale d'un centre d'appels dans un environnement "blend". Département d'informatique et de recherche opérationnelle, Université de Montréal, Janvier 2002.
- [22] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. GNU Scientific Library version 1.2 : Reference manual, Juillet 2002. http://sources.redhat.com/gsl/ref/gsl-ref_toc.html.
- [23] N. Gans and Y.-P. Zhou. A call-routing problem with service-level constraints. Technical report, The Wharton School, University of Pennsylvania, Philadelphia, September 2001. Preprint.
- [24] L. Green and P.J. Kolesar. The pointwise stationnary approximation for queues with nonstationary arrivals. *Management Science*, 37(1) :84–97, 1991.

- [25] L. Green, P.J. Kolesar, and A. Svoronos. Some effects of nonstationarity on multiserver markovian queueing systems. *Operations Research*, 39(3) :502–511, 1991.
- [26] S. Gulati. Call center scheduling technology evaluation using simulation. In *Proceedings of the 2001 Winter Simulation Conference*, volume 2, pages 1438–1442, 2001.
- [27] S. Henderson and A. Mason. Rostering by iterating integer programming and simulation. In *Proceedings of the 1998 Winter Simulation Conference*, volume 1, pages 677–683, 1998.
- [28] I. D. Hill, R. Hill, and R. L. Holder. Fitting Johnson curves by moments. *Applied Statistics*, 25(2) :180–189, 1976.
- [29] O. B. Jennings, A. Mandelbaum, W. A. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Science*, 42(10) :1383–1394, 1996.
- [30] N. L. Johnson and S. Kotz. *Discrete distribution*. Wiley, 1972.
- [31] G. Jongbloed and G. Koole. Managing uncertainty in call centers using Poisson mixtures. *Applied Stochastic Models in Business and Industry*, 17(4) :307–318, 2001.
- [32] T. A. Grossman Jr., S. L. Oh, T. R. Rohleder, and D. A. Samuelson. *The Encyclopedia of Operations Research and Management Science*. Kluwer Academic Publishers, second edition, 2000.
- [33] R. Klungle. Simulation of a claims call center : a success and a failure. In *Proceedings of the 1999 Winter Simulation Conference*, volume 2, pages 1648–1652, 1999.
- [34] D. E. Knuth. *The Art of Computer Programming, Volume 2 : Seminumerical Algorithms*. Addison-Wesley, Reading, Mass., third edition, 1998.
- [35] G. Koole. Mathematical modeling of call centers. Technical report, Vrije Universiteit, Amsterdam, 2001.
- [36] A. Lakhany and H. Mausser. Estimating the parameters of the generalized lambda distribution. *Algo research quarterly*, 3(3) :47–58, 2000.
- [37] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, third edition, 2000.
- [38] P. L’Ecuyer. *SIMOD : Définition fonctionnelle et guide d’utilisation*, Jan. 2000.
- [39] P. L’Ecuyer. *SSJ : A Java Library for Stochastic Simulation*, 2001. Software user’s guide.
- [40] P. L’Ecuyer. *SSC : A Library for Stochastic Simulation in C*, 2002. Software user’s guide.

- [41] P. L'Ecuyer, L. Meliani, and J. Vaucher. SSJ : A framework for stochastic simulation in Java. In *Proceedings of the 2002 Winter Simulation Conference*, 2002. to appear.
- [42] J. Leydold, W. Hörmann, E. Janka, and G. Tirlir. UNURAND : Universal Non-Uniform RANdom number generators, 2002. UNURAND User Manual, <http://statistik.wu-wien.ac.at/unuran/doc.html>.
- [43] MATLAB. *MATLAB Reference Manual*. The MathWorks Inc., Natick, Mass., 1992.
- [44] A. A. B. Pritsker. Principles of simulation modeling. In Jerry Banks, editor, *Handbook of Simulation*, pages 31–51. Wiley, 1998. chapter 2.
- [45] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, California, second edition, 1995.
- [46] J. Riordan. *Stochastic service systems*. John Wiley & Sons Inc., New York, 1962. The SIAM series in applied mathematics.
- [47] C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, 1999.
- [48] S. M. Ross. *Simulation*. Academic Press, second edition, 1997.
- [49] R. M. Saltzman and V. Mehrotra. A call center uses simulation to drive strategic change. *Interfaces*, 31(3) :87–101, 2001.
- [50] J. J. Swain, S. Venkatraman, and J. R. Wilson. Least-squares estimation of distribution functions in Johnson's translation system. *Journal of Statistical Computation and Simulation*, 29 :271–297, 1988.
- [51] O. Tanir and R. J. Booth. Call center simulation in *Bell Canada*. In *Proceedings of the 1999 Winter Simulation Conference*, volume 2, pages 1640–1647, 1999.
- [52] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, San Diego, third edition, 1998.
- [53] N. M. van Dijk. On hybrid combination of queueing and simulation. In *Proceedings of the 2000 Winter Simulation Conference*, volume 1, pages 147–150, 2000.
- [54] R. E. Wheeler. Quantiles estimators of Johnson curve parameters. *Biometrika*, 63(3) :725–728, 1980.
- [55] W. Whitt. The pointwise stationary approximation for $M(t)/M(t)/s$ queues is asymptotically correct as the rates increase. *Management Science*, 37(3) :307–314, 1991.
- [56] W. Whitt. Dynamic staffing in a telephone call center aiming to immediately answer all calls. *Operations Research Letters*, 24 :205–212, 1999.

- [57] R. W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice-Hall, New York, 1989.

ANNEXE

Librairie de simulation pour un centre d'appels en mode mixte

Guide d'utilisation

(Version : 1 mars 2003)

Alexandre Deslauriers

Introduction

Dans ce guide, il est question de la librairie utilisée pour la simulation d'un modèle représentant un centre d'appels en mode mixte. Le but de cette librairie est d'estimer des mesures de performance pour ce type de modèles. Ces dernières seront obtenues en simulant le modèle sur plusieurs jours.

La structure de la librairie est fixe. Elle est divisée en différents modules qui représentent chacun un aspect précis du centre d'appels. Chaque module a un fonctionnement interne indépendant, mais il y a des interrelations possibles entre les modules. Voici la liste des modules ainsi que leur rôle dans la simulation :

- Module **Incall** : Représente les appels entrants. Son principal rôle est de gérer les arrivées d'appels. Il recueille aussi des mesures de performance sur les appels entrants.
- Module **Servers** : Représente les agents dans le centre d'appels. Il gère la disponibilité des agents et la durée de service des appels entrants et sortants.
- Module **Waitqueue** : Représente la file d'attente. Il gère les priorités dans la file ainsi que les abandons.
- Module **Dialer** : Représente le système de composition d'appels sortants. Ce module décide du nombre d'appels à composer en se basant sur diverses informations sur le système. Celles-ci sont généralement obtenues en questionnant les autres modules.
- Module **Outcall** : Représente les appels sortants. Ce module fait le lien entre le système de composition d'appels sortants et les agents pouvant servir ce type d'appels.
- Module **Statistic** : Recueille, tout au long d'une simulation, des mesures de performance journalières sur le modèle simulé.
- Module **Distribution** : Permet la génération de variables aléatoires suivant des lois de probabilité particulières et de processus stochastiques non-conventionnels.

- Module **Constant** : Contient les paramètres généraux à tous les modules dont les heures d'ouverture et de fermeture.

- Module **Call** : Peut être considéré comme le programme principal de nos simulations. C'est lui qui initialise tous les modules à chaque début de journée et qui simule le modèle sur plusieurs jours.

Ce guide présente une description de ces modules et des interrelations possibles entre eux. Une interrelation est définie par une fonction visible par les autres modules et elle peut être appelée en tout temps.

La structure générale du modèle à simuler est fixée par les modules. Le changement d'une partie de cette structure est possible en modifiant le fonctionnement interne du module qui lui est associé. Les changements de paramètres du modèle sont possibles en modifiant le fichier de paramètres d'entrée.

Module Distribution

Ce module permet de générer des variables aléatoires suivant des lois ou des processus stochastiques non-conventionnels. Les lois définies ont souvent la particularité d'avoir des paramètres dépendant de l'instant de la journée. Ainsi, la génération de ces variables doit se faire en s'assurant d'avoir les paramètres courants. Il est possible de générer des variables de loi Bernouilli, exponentielle, lognormale, gamma et Weibull.

Les autres processus que l'on retrouve dans ce module touchent la modélisation des arrivées. Il est possible de générer les instants d'arrivée avec les processus d'arrivée étudiés dans le mémoire. Les possibilités sont un processus de type Poisson-Gamma et de Poisson non homogène avec ou sans corrélation.

Nous donnons aussi la possibilité de générer des variables aléatoires suivant une distribution empirique construite par noyau. Dans ce cas, nous utilisons les méthodes de la librairie UNURAN [42].

La construction de ce module a été nécessaire dans les cas où nous ne pouvions pas utiliser directement les fonctions de génération de SSC [40]. La non stationnarité des paramètres et la complexité de nos modèles nous ont forcé à développer ces méthodes.

Pour chaque loi et processus, nous devons créer un objet associé en utilisant la fonction de création et en donnant un lien à un fichier contenant ces paramètres. Les paramètres débutent après la lecture du signe `$$`. Chaque ligne doit contenir les paramètres d'un intervalle de temps ainsi que la durée de cet intervalle. Le fichier pour la distribution empirique doit contenir l'ensemble des données (une par ligne). Ensuite, on peut générer les variables en appelant les fonctions `Distribution_LawReturnIntGenerationFunction` pour une variable discrète et `Distribution_LawReturnDoubleGenerationFunction` pour une variable continue, en spécifiant l'instant de génération et l'objet associé à la loi ou processus.

```
typedef struct Distribution_LawReturnInt_ * Distribution_LawReturnInt;
struct Distribution_LawReturnInt_
{
    int (*GenerationFunction) (Distribution_LawReturnInt Law, double SimTime);
};
```

L'objet `Distribution_LawReturnInt_` permet de générer une variable aléatoire discrète dont la loi a des paramètres variant dans le temps. `GenerationFunction` pointe sur la bonne fonction

de génération à utiliser.

```
typedef struct Distribution_LawReturnDouble_ * Distribution_LawReturnDouble;
struct Distribution_LawReturnDouble_
{
    double (*GenerationFunction) (Distribution_LawReturnDouble Law, double SimTime);
};
```

L'objet `Distribution_LawReturnDouble_` permet de générer une variable aléatoire continue dont la loi a des paramètres variant dans le temps. `GenerationFunction` pointe sur la bonne fonction de génération à utiliser.

```
typedef void * Distribution_Law;
```

Objet général représentant autant un objet `Distribution_LawReturnInt_` qu'un objet `Distribution_LawReturnDouble_`.

```
typedef Distribution_LawReturnInt Distribution_Bernouilli;
```

Définit une loi Bernouilli dont les paramètres varient selon l'instant de la journée. La génération d'une telle variable est expliquée dans le mémoire.

```
typedef Distribution_LawReturnDouble Distribution_Exponential;
```

Définit une loi exponentielle dont les paramètres varient selon l'instant de la journée. La génération pour un instant précis utilise la fonction de SSC (méthode d'inversion) [40].

```
typedef Distribution_LawReturnDouble Distribution_Gamma;
```

Définit une loi gamma dont les paramètres varient selon l'instant de la journée. La génération pour un instant précis utilise la fonction de SSC [40].

```
typedef Distribution_LawReturnDouble Distribution_Weibull;
```

Définit une loi Weibull dont les paramètres varient selon l'instant de la journée. La génération pour un instant précis utilise la fonction de SSC [40].

```
typedef Distribution_LawReturnDouble Distribution_Lognormal;
```

Définit une loi lognormale dont les paramètres varient selon l'instant de la journée. Sa génération utilise la fonction de SSC (méthode d'inversion) [40].

```
typedef Distribution_LawReturnDouble Distribution_PoissonNonHomoCorrelated;
```

Définit un processus d'arrivée doublement stochastique décrit dans [56]. Ce processus a un taux d'arrivée stochastique $\Lambda(t)$ de la forme :

$$\Lambda(t) = W\lambda(t), t \geq 0 \quad (5.2)$$

où $\lambda(t) = \lambda_i$ lorsque t est dans la demi-heure i et W est une variable aléatoire positive associée à une journée. Nous supposons que W est de loi gamma. Une fois le $W = w$ fixé, nous avons un processus de Poisson non-homogène de taux $w\lambda(t)$. Il est important de noter que nous représentons un processus de Poisson non-homogène lorsque les deux paramètres de la loi gamma sont égaux à 0. Les détails et la méthode de génération du processus sont expliqués dans le mémoire.

```
typedef Distribution_LawReturnDouble Distribution_PoissonGamma;
```

Définit un processus d'arrivée de type Poisson-Gamma décrit dans [31] dont les paramètres varient selon l'instant de la journée. Les détails et la méthode de génération du processus sont expliqués dans le mémoire.

```
typedef Distribution_LawReturnDouble Distribution_Constant;
```

Définit une constante dont la valeur varie selon l'instant de la journée.

```
typedef Distribution_LawReturnDouble Distribution_UNURAN;
```

Définit une variable aléatoire suivant une distribution empirique construite par noyau. La méthode de construction est décrite dans le guide de la librairie UNURAN [42].

```
Distribution_Bernouilli Distribution_CreateBernouilli (FILE * f);
```

Retourne et construit un objet représentant une loi Bernouilli dont les paramètres variables au cours d'une journée sont détaillés dans le fichier `f`.

```
Distribution_Exponential Distribution_CreateExponential (FILE * f);
```

Retourne et construit un objet représentant une loi exponentielle dont les paramètres variables au cours d'une journée sont détaillés dans le fichier `f`.

```
Distribution_Gamma Distribution_CreateGamma (FILE * f);
```

Retourne et construit un objet représentant une loi gamma dont les paramètres variables au cours d'une journée sont détaillés dans le fichier `f`.

```
Distribution_Gamma Distribution_CreateWeibull (FILE * f);
```

Retourne et construit un objet représentant une loi Weibull dont les paramètres variables au cours d'une journée sont détaillés dans le fichier `f`.

```
Distribution_Lognormal Distribution_CreateLognormal (FILE * f);
```

Retourne et construit un objet représentant une loi lognormale dont les paramètres variables au cours d'une journée sont détaillés dans le fichier `f`.

```
Distribution_PoissonNonHomoCorrelated Distribution_CreatePoissonNonHomoCorrelated  
(FILE * f);
```

Retourne et construit un objet représentant un processus de type `Distribution_PoissonNonHomoCorrelated`. Ses paramètres sont détaillés dans le fichier `f`.

```
Distribution_PoissonGamma Distribution_CreatePoissonGamma (FILE * f);
```

Retourne et construit un objet représentant un processus de type `Distribution_PoissonGamma`. Ses paramètres sont détaillés dans le fichier `f`.

```
Distribution_Constant Distribution_CreateConstant (FILE * f);
```

Retourne et construit un objet représentant une constante variant au cours de la journée. Ses valeurs sont détaillées dans le fichier `f`.

```
Distribution_UNURAN Distribution_CreateUNURAN (const char * file);
```

Retourne et construit un objet représentant une variable aléatoire suivant une distribution empirique construite par noyau. La distribution est construite à partir des observations individuelles contenues dans le fichier décrit par la chaîne de caractères `file`.

```
int Distribution_LawReturnIntGenerationFunction (Distribution_LawReturnInt L,  
double SimTime);
```

Génère une valeur aléatoire selon la loi discrète `L` et l'instant `SimTime`.

```
double Distribution_LawReturnDoubleGenerationFunction (Distribution_LawReturnDouble  
L, double SimTime);
```

Génère une valeur aléatoire selon la loi continue ou le processus `L` et l'instant `SimTime`.

```
void Distribution_InitLawReturnInt (Distribution_LawReturnInt Law);
```

Initialise la loi discrète `Law` pour le début d'une journée.

```
void Distribution_InitLawReturnDouble (Distribution_LawReturnDouble Law);
```

Initialise la loi continue ou le processus `Law` pour le début d'une journée.

Module Incall

Incall est le module qui représente les appels entrants. On active ce module en appelant sa fonction de création qui reçoit en paramètre un tableau de lois et de processus. Ces derniers modélisent les aspects touchant ce type appel.

```
void Incall_Create (Distribution_Law P []);
```

Crée et active le module **Incall**. Le tableau **P** contient les lois associées aux appels entrants. Sa première case contient le processus d'arrivée des appels et la deuxième contient la loi de la durée qu'un client entrant est prêt à attendre dans la file. Ces lois seront utilisées uniquement par ce module. Il est possible d'ajouter des nouveaux aspects aléatoires en utilisant d'autres cases du tableau **P**.

```
void Incall_Init (void);
```

Cette fonction initialise le module pour le début d'une nouvelle journée dans le centre d'appels. Les mesures journalières sont remises à zéro et les paramètres de ses lois sont ré-initialisés.

```
void Incall_BeginCollectStat (void);
```

Avertit le module qu'il doit commencer à recueillir des mesures d'intérêt sur les appels entrants. Il s'agit du nombre d'appels arrivés, perdus, servis, ayant attendu plus de 20 secondes, etc.

```
void Incall_NewDone (struct InfoCall * call);
```

Cette fonction permet d'avertir le module **Incall** que le système a servi un appel entrant supplémentaire. En général, c'est le module **Servers** qui utilise cette fonction à la fin du service d'un appel.

```
void Incall_NewLost (struct InfoCall * call);
```

Cette fonction permet d'avertir le module **Incall** que le système a obtenu un abandon supplémentaire. En général, c'est le module **Waitqueue** qui utilise cette fonction.

```
double Incall_MaxWait (void);
```

Cette fonction génère le temps qu'un client est prêt à attendre dans la file d'attente. La loi représentant cet aspect a été définie à la création du module **Incall**.

```
void Incall_ArrivalScheduling (void);
```

Cette fonction planifie l'arrivée d'un nouvel appel entrant. Le processus d'arrivée a été défini à la création du module **Incall**. On appelle cette fonction au début de la journée et à chaque instant d'arrivée d'un appel (pour planifier le prochain). On informe le module **Servers** dès qu'on a un

nouvel arrivée.

```
void Incall_ArrivalStopping (void);
```

Cette fonction arrête les arrivées d'appels entrants. On appelle cette fonction à la fermeture du centre.

```
void Incall_GeneratingWaitBy (struct InfoCall * call);
```

À chaque début de service d'un client entrant, le module `Servers` informe, à l'aide de cette fonction, le module `Incall` que l'appel entrant `call` n'aura plus à attendre. Ainsi, il peut calculer l'attente de cet appel et mettre à jour les mesures de performance touchant l'attente des clients entrants.

```
int Incall_NbInDone (void);
```

Retourne le nombre d'appels entrants servis depuis le début de la journée courante.

```
int Incall_NbInLost (void);
```

Retourne le nombre d'abandons (appels entrants) depuis le début de la journée courante.

```
int Incall_NbInMore20 (void);
```

Retourne le nombre d'appels entrants reçus (servis et abandons) ayant attendu plus de 20 secondes dans la file depuis le début de la journée courante.

```
int Incall_NbInMore30 (void);
```

Retourne le nombre d'appels entrants reçus (servis et abandons) ayant attendu plus de 30 secondes dans la file depuis le début de la journée courante.

```
int Incall_NbInWaiting (void);
```

Retourne le nombre d'appels entrants ayant attendu dans la file depuis le début de la journée courante.

```
int Incall_TotalInWaitingTime (void);
```

Retourne l'attente totale générée par les appels entrants ayant attendu dans la file depuis le début de la journée courante.

```
double Incall_QualofServ (void);
```

Retourne un indice de qualité de service sur les appels entrants. La version courante de la librairie retourne la proportion d'appels entrants ayant attendu plus de 20 secondes durant les 10 dernières minutes. Cette mesure est utilisée par le module `Dialer` pour savoir s'il doit composer de nouveaux appels sortants.

```
void Incall_ReportDH (int NRep) ;
```

Construit un programme S-plus permettant d'analyser (sous forme de graphique) le pourcentage d'appels ayant attendu moins de 20 secondes et le pourcentage d'abandons sur chaque demi-heure de la journée.

Module Servers

Le module **Servers** représente le service des appels entrants et sortants. Il modélise les durées de service et les deux groupes d'agents : groupe I servant uniquement les appels entrants et groupe B servant les deux types d'appels. Ces aspects sont des variables aléatoires que l'on définit à la création du modèle.

```
void Servers_CreateGroup (Distribution_Law P []);
```

Crée et active le module **Servers**. Le tableau **P** contient les lois associées au service des appels. Les deux premières cases contiennent respectivement la loi de la durée de service d'un appel entrant et celle d'un appel sortant. Les deux cases suivantes représentent la disponibilité des agents dans les deux groupes. Ces lois seront utilisées uniquement par ce module. Il est possible d'ajouter des nouveaux aspects aléatoires avec d'autres cases du tableau **P**.

```
void Servers_InitServersGroup (void);
```

Initialise le module pour le début d'une nouvelle journée dans le centre d'appels. Les mesures journalières sont remises à zéro et les paramètres de ses lois sont ré-initialisés.

```
void Servers_BeginCollectStat(void);
```

Avertit le module qu'il doit commencer à recueillir des mesures d'intérêt sur le service des appels. Il s'agit du volume d'appels entrants servis, du volume d'appels sortants servis, du taux d'occupation des agents, etc.

```
void Servers_Type1ServiceTraitement (struct InfoCall * call);
```

Avertit le module qu'on a un nouvel appel entrant **call** à servir. Cet appel peut venir de l'extérieur (module **Incall**) ou de la file d'attente (module **Waitqueue**). Avant tout, on vérifie s'il est possible de le servir, i.e. on vérifie le nombre d'agents libres dans les deux groupes. S'il y en a au moins un de libre alors on génère sa durée de service (définie par le tableau **P** à la création du module) et on occupe un nouvel agent du groupe I si possible (on prend un agent du groupe B sinon). Une fois que le service de l'appel débute, nous informons le module **Incall** de son attente dans la file car l'appel n'a plus à attendre. Si tous les agents sont occupés alors on insère l'appel dans la file d'attente et on doit communiquer avec le module **Waitqueue**.

```
void Servers_Type2ServiceTraitement (struct InfoCall * call);
```

Avertit le module qu'on a un nouvel appel **call** sortant à servir. Cet appel vient du système de composition d'appels sortants (module **Dialer**). Ce dernier a réussi à rejoindre un client sortant et demande au groupe d'agents s'il est possible de le servir. Donc, on vérifie le nombre d'agents

libres dans le groupe B. S'il y en a au moins un de libre alors on génère sa durée de service (définie par le tableau P à la création du module) et on occupe un nouvel agent du groupe B. Si tous les agents du groupe B sont occupés alors nous avons un nouveau "mismatch" que l'on communique au module Dialer.

```
void Servers_Type1EndService (void);
```

Avertit le module de la fin de service d'un appel entrant. On libère l'agent qui servait cet appel et on met à jour certaines mesures d'intérêts : volume d'appels entrants, nombre d'appels entrants servis (module Incall). Ensuite, on communique avec le module Waitqueue pour savoir s'il y a un appel en attente pouvant se faire servir par le nouvel agent libre. Et finalement, on demande au module Dialer de prendre une décision sur le nombre d'appels sortants à composer.

```
void Servers_Type2EndService (void);
```

Avertit le module de la fin de service d'un appel sortant. On libère l'agent du groupe B qui servait cet appel et on met à jour certaines mesures d'intérêts : volume d'appels sortants, nombre d'appels sortants servis (module Outcall). Ensuite, on communique avec le module Waitqueue pour savoir s'il y a un appel en attente pouvant se faire servir par le nouvel agent libre. Et finalement, on demande au module Dialer de prendre une décision sur le nombre d'appels sortants à composer.

```
int Servers_NbAgentsAvailTypeI (void);
```

Retourne le nombre d'agents disponibles dans le groupe d'agents I. Ce groupe ne sert que les appels entrants.

```
int Servers_NbAgentsAvailTypeB (void);
```

Retourne le nombre d'agents disponibles dans le groupe d'agents B. Ce groupe sert les deux types d'appels (entrants et sortants).

```
int Servers_NbAgentsAvail (void);
```

Retourne le nombre d'agents disponibles dans les deux groupes d'agents.

```
int Servers_NbAgentsFreeTypeI (void);
```

Retourne le nombre d'agents libres (en attente) dans le groupe d'agents I.

```
int Servers_NbAgentsFreeTypeB (void);
```

Retourne le nombre d'agents libres (en attente) dans le groupe d'agents B.

```
int Servers_NbAgentsFree (void);
```

Retourne le nombre d'agents libres (en attente) dans les deux groupes d'agents.

```
int Servers_NbAgentsBusyTypeI (void);
```

Retourne le nombre d'agents occupés (en service) dans le groupe d'agents I.

```
int Servers_NbAgentsBusyTypeB (void);
```

Retourne le nombre d'agents occupés (en service) dans le groupe d'agents B.

```
int Servers_NbAgentsBusy (void);
```

Retourne le nombre d'agents occupés (en service) dans les deux groupes.

```
double Servers_OccupancyMean (void);
```

Retourne le taux d'occupation des agents des 2 groupes depuis le début de la simulation.

```
double Servers_OccupancyMeanDay (void);
```

Retourne le taux d'occupation des agents des 2 groupes depuis le début de la journée courante.

```
void Servers_OccupancyMeanReport (int NRep);
```

Construit un programme S-plus permettant d'analyser (sous forme de graphique) le volume d'appels entrants servis, le volume d'appels sortants servis et le taux d'occupation sur chaque demi-heure de la journée.

Module Waitqueue

Le module `Waitqueue` représente la file d'attente du centre d'appels. La règle de priorité pour les appels entrants est du type premier arrivée - premier servi (FIFO : First In First Out). Il n'y a qu'une seule file pour les appels entrants et sortants, mais les appels sortants ont priorité sur les appels entrants. La file a une capacité infinie. On active le module en appelant la fonction de création.

```
void Waitqueue_CreateQueue (void);
```

Crée et active le module `Waitqueue`. On construit la file d'attente qui est une liste doublement chaînée (module `List` de SSC [40]) acceptant des objets `InfoCall` (voir module `Constant`).

```
void Waitqueue_InitQueue (void);
```

Initialise la file d'attente pour le début d'une journée. On vide la file.

```
void Waitqueue_BeginCollectStat(void);
```

Avertit le module qu'il doit commencer à recueillir les mesures d'intérêt (journalières) sur la file d'attente. On recueille l'attente moyenne, le nombre moyen de clients dans la file, etc.

```
void Waitqueue_Insert (struct InfoCall * call);
```

Insère l'appel `call` dans la file d'attente selon sa priorité. Avant de l'insérer, on calcule le temps maximal que le client est prêt à attendre dans la file. Cette durée dépend du type d'appels (module `Incall` ou `Outcall`).

```
void Waitqueue_NextCall (void);
```

Cette fonction demande au module de vérifier la possibilité de servir un client en attente. Tout d'abord, on doit trouver le premier appel dans la file qui est toujours en attente. Si la file est vide alors on ne fait rien. Sinon, on prend ce premier appel et on tente de le servir en communiquant avec le module `Servers`.

```
int Waitqueue_NbInQueue (void);
```

Retourne le nombre courant de clients dans la file d'attente.

```
double Waitqueue_AverageNbInQueue (void);
```

Retourne le nombre moyen de clients (par unité de temps) dans la file depuis le début de la journée.

```
double Waitqueue_CumulateWait(void);
```

Retourne l'attente totale dans la file depuis le début de la journée.

```
double Waitqueue_AverageSojournTime (void);
```

Retourne l'attente moyenne dans la file depuis le début de la journée.

Module Dialer

Ce module représente le système de composition d'appels sortants. Celui-ci décide du nombre d'appels sortants à composer en se basant sur différentes informations (qualité de service, occupation des agents, ...). On active le module en appelant la fonction de création.

```
void Dialer_CreateDialer (Distribution_Law P []);
```

Crée et active le module `Dialer`. Le tableau `P` contient les lois associées à la composition des appels sortants. Sa première case représente la probabilité de rejoindre un client par un appel sortant. La deuxième et troisième représente respectivement le temps nécessaire pour rejoindre un client et pour se rendre compte que le client ne pourra pas être rejoint. Ces lois seront utilisées uniquement par ce module. Il est possible d'ajouter d'autres aspects aléatoires avec d'autres cases du tableau `P`.

```
void Dialer_InitDialer (void);
```

Initialise le module pour le début d'une nouvelle journée dans le centre d'appels. Les mesures journalières sont mises à zéro et les paramètres des lois sont ré-initialisés.

```
void Dialer_BeginCollectStat(void);
```

Avertit le module qu'il doit commencer à recueillir les mesures d'intérêt (journalières) sur la composition d'appels sortants. Il s'agit du nombre de "mismatches", du nombre d'appels composés, etc.

```
void Dialer_Compose (void);
```

Demande au module de composer (si nécessaire) de nouveaux appels sortants. Le nombre à composer dépend d'un algorithme de décision spécifique. En ce moment, il utilise l'algorithme de composition décrit dans le mémoire, i.e. celui qui permet d'obtenir des mesures assez près de ce qu'on retrouve dans le centre d'appels Contact Unique. Voici son algorithme : Tout d'abord, on vérifie les trois conditions suivantes : si le nombre d'agents libres dans le groupe B (`Servers_NbAgentsFreeType2`) est supérieur à 2, si le nombre total d'agents libres dans les deux groupes (`Servers_NbAgentsFree`) est supérieur à 4 et si plus de 75 % des appels entrants (agrégés par minute) depuis les 10 dernières minutes ont attendu moins de 20 secondes (`Incall_QualofServ`). Si ces trois conditions ne sont pas toutes satisfaites, on n'effectue aucune composition d'appel. Dans le cas contraire, le nombre d'appels sortants composés est égal à deux fois le nombre d'agents libres dans le groupe B si le pourcentage de "mismatches" (agrégés par minute) depuis les 15 dernières minutes ne dépasse 8% (`Dialer_QualofServ`) et à seulement une fois le nombre d'agents libres dans le groupe B sinon. Les appels sont composés en parallèle et on

décide du nombre de clients ayant été rejoints à l'aide de la loi définie à l'activation du module. Pour chaque appel rejoint, on informe le module `Outcall` pour qu'il se charge du traitement de son service.

```
void Dialer_NewMismatch (struct InfoCall * call);
```

Informe le module sur le fait que l'appel `call` est un "mismatch". On insère cet appel dans la file d'attente (module `Waitqueue`) si le client est prêt à attendre avant de se faire servir.

```
int Dialer_NbCompose (void);
```

Retourne le nombre d'appels que le système a composé depuis le début de la journée.

```
int Dialer_NbReach (void);
```

Retourne le nombre de clients que le système a rejoint depuis le début de la journée.

```
int Dialer_NbMismatch (void);
```

Retourne le nombre de "mismatches" que le système a fait depuis le début de la journée.

Module Outcall

Le module `Outcall` représente les appels sortants. Il s'occupe de générer le temps que ce type d'appels est prêt à attendre dans la file et de recueillir des mesures de performance sur les appels sortants. On active le module en appelant sa fonction de création.

```
void Outcall_Create (Distribution_Law P []);
```

Crée et active le module `Outcall`. Le tableau `P` contient la loi associée à la durée qu'un client d'un appel sortant est prêt à attendre dans la file. Il est possible d'ajouter d'autres aspects aléatoires avec d'autres cases du tableau `P`.

```
void Outcall_Init (void);
```

Initialise la loi et les mesures du module pour le début d'une journée.

```
void Outcall_BeginCollectStat(void);
```

Avertit le module qu'il doit commencer à recueillir les mesures d'intérêt (journalières) sur les appels sortants. Il s'agit du nombre d'appels servis, perdus et de l'attente générée.

```
void Outcall_NewDone (struct InfoCall * call);
```

Avertit le module que l'appel `call` vient d'obtenir son service. En général, c'est le module `Servers` qui utilise cette fonction à la fin du service d'un appel.

```
void Outcall_NewLost (struct InfoCall * call);
```

Avertit le module que l'appel `call` vient d'abandonner le système sans avoir reçu son service. En général, c'est le module `Waitqueue` qui utilise cette fonction.

```
double Outcall_MaxWait (void);
```

Génère la durée qu'un client d'un appel sortant est prêt à attendre dans la file. La loi représentant cet aspect a été définie à la création du module `Outcall` (tableau `P`).

```
void Outcall_ArrivalScheduling (void);
```

Cette fonction avertit le module qu'il y a un nouvel appel sortant à servir. La fonction est utilisée par le module `Dialer` lorsqu'il a réussi à rejoindre un client à l'aide de la composition d'un appel sortant. Il y a mise à jour des mesures et ensuite, le module `Servers` se fait avertir qu'il y a un nouvel appel sortant à servir (`Servers_Type2ServiceTraitement`).

```
void Outcall_GeneratingWaitBy (struct InfoCall * call);
```

À chaque début de service d'un client sortant, le module `Servers` informe (par cette fonction) le module `Outcall` que l'appel sortant `call` n'aura plus à attendre. Ainsi, il peut calculer l'attente

de cet appel et mettre à jour ses mesures de performance touchant l'attente des clients sortants.

```
int Outcall_NbOutDone (void);
```

Retourne le nombre d'appels sortants servis depuis le début de la journée courante.

```
int Outcall_NbOutLost (void);
```

Retourne le nombre d'appels sortants perdus depuis le début de la journée courante.

Module **Statistic**

Ce module cumule un ensemble de mesures journalières sur le centre d'appels. À la fin de chaque journée simulée, on met à jour des blocs statistiques définis dans la librairie SSC [40]. Ces mises à jour permettent de calculer une moyenne, un écart-type, un minimum, un maximum et un intervalle de confiance sur plusieurs aspects du modèle représentant un centre d'appels. Les mesures journalières considèrent le nombre d'appels entrants servis, d'appels entrants perdus, d'appels sortants servis, d'appels sortants composés et de "mismatches". Nous nous intéressons aussi à l'attente des clients : nous calculons la durée moyenne d'attente, la durée moyenne d'attente sachant qu'un client doit attendre, le nombre moyen de clients dans la file par unité de temps et le pourcentage de clients ayant attendu moins de 20 (et 30) secondes. Et finalement, on présente une mesure sur l'occupation des agents tout au long de la simulation.

Pour utiliser le module, il suffit d'activer le module et d'appeler la fonction de mise à jour à la fin de chaque journée simulée. Cette fonction fait la mise à jour en obtenant diverses informations sur les autres modules.

Il est possible d'obtenir un rapport détaillé sur les mesures d'intérêt. Pour chacune d'entre elles, nous présentons sa moyenne, son écart-type, son minimum, son maximum et un intervalle de confiance.

```
void Statistic_CreateUnit (char * Descriptor) ;
```

Crée et active le module **Statistic**. La chaîne de caractères **Descriptor** contient une description de l'ensemble des mesures recueillies. Cette fonction construit les blocs statistiques nécessaires aux calculs des mesures.

```
void Statistic_InitUnit (void) ;
```

Initialise le module pour le début d'une simulation (et non une journée). Les blocs statistiques sont mis à zéro et on calcule des mesures à partir de cette initialisation.

```
void Statistic_Update (void) ;
```

Met à jour les blocs statistiques du module à la fin d'une journée. Cette mise à jour est possible en questionnant les autres modules sur les mesures recueillies pour cette journée.

```
void Statistic_Report (long NbRep) ;
```

Imprime un rapport sur les mesures calculées depuis la dernière initialisation. Pour chaque mesure, on présente sa moyenne, son écart-type, son minimum et son maximum.

```
void Statistic_Report2 (long NbRep) ;
```

Imprime un rapport sur les mesures calculées depuis la dernière initialisation. Pour chaque mesure, on présente sa moyenne, son minimum et son maximum et un intervalle de confiance (95%)

Exemple d'un rapport fourni par ce module :

```
Resultats de simulation pour      10000 repetitions
```

```
Periode etudiee = statistiques journalieres
```

	Min	Max	Mean	IntCon	Rep
Nbr d'appels sortants servis par jour	121.0	741.0	511.9	1.7	10000
Nbr d'appels entrants servis par jour	530.0	1797.0	1169.0	3.8	10000
Nbr d'appels entrants perdus par jour	0.0	452.0	31.8	0.8	10000
Nbr d'appels entrants servis et perdus par jour	531.0	2243.0	1200.8	4.5	10000
Moyenne de la duree d'attente (sec)	--	--	12.8	0.3	10000
Moyenne de la duree d'attente sachant que le client a attendu par jour (sec)	--	--	71.8	0.7	10000
Nb. dans la queue par unite de temps par jour	0.0	5.2	0.4	0.0	10000
Nbr dont attente > 20 sec par jour	0.0	1435.0	164.1	3.6	10000
Nbr dont attente > 30 sec par jour	0.0	1347.0	144.8	3.3	10000
Nbr de mismatch par jour	5.0	73.0	37.6	0.2	10000

Nbr d'appels sortants composes par jour	325.0	2421.0	1666.0	5.5	10000
Taux d'occupation			71.5	0.1	
Pourcentage des clients entrants dont attente < 20 sec	--	--	86.3	0.0	10000
Pourcentage des clients entrants dont attente < 30 sec	--	--	87.9	0.0	10000

Module Constant

Ce module comprend un ensemble de constantes utilisées dans plusieurs modules de la librairie.

```
#define INF 10E9

#define Hour 1.0 /* One hour */
#define Minute (1.0 / 60.0) /* One minute */
#define Second (1.0 / 3600.0) /* One second */
```

L'unité de temps utilisé dans la librairie est l'heure. Les constantes `Minute` et `Second` permettent de convertir un autre unité de temps (ici les minutes et les secondes) en heures.

```
typedef enum {In, Out} TypeOfCall;
/* Type d'appel */

typedef enum {InOnly, Blend} TypeOfServ;
/* Type de serveur */

struct InfoCall
/* Information sur l'appel (ou client) */
{
    TypeOfCall TypeCall; /* Son type */
    double ArrivTime; /* Son instant d'arrivee */
    double MaxWait; /* Le temps d'attente maximal du
    client dans la file */
    double ServTime; /* Temps de service */
    TypeOfServ TypeServ; /* Type de serveur */
};
```

Plusieurs informations sur les appels (ou clients) sont essentielles à leur traitement dans la simulation. Nous avons défini cet objet qui englobe les informations importantes. On emmagasine son type (entrant ou sortant), son type de serveur (groupe I ou B), son instant d'arrivée, la durée qu'il est prêt à attendre dans la file et la durée de son service.

```
void Constant_CreateConstEv (FILE * f);
```

Active le module avec le fichier de paramètres `f`. La première ligne de paramètres débute après le symbole `$$` et elle décrit l'instant du début des arrivées d'appels entrants, l'instant d'ouverture du centre, l'instant du début de la composition des appels sortants et l'instant de fermeture du centre.

```
double Constant_InitDay (void);
```

Retourne l'instant du début des arrivées d'appels entrants.

double Constant_StartDay (void) ;

Retourne l'instant de l'ouverture du centre, i.e. les serveurs peuvent servir les clients dans le système.

double Constant_StartDialer (void) ;

Retourne l'instant du début de la composition des appels sortants.

double Constant_EndDay (void) ;

Retourne l'instant de la fermeture du centre. Après cet instant, il n'y a plus d'arrivées d'appels entrants et de composition d'appels sortants. Les agents terminent le service des appels déjà dans le système.

Module Call

Ce module peut être considéré comme le programme principal de la simulation. C'est lui qui active les modules (fixe leurs différentes lois) et les initialise à chaque début de journée. Il permet de calculer des mesures de performance sur le modèle à l'aide de la simulation de plusieurs jours indépendants.

Avant de faire nos simulations, on doit construire les tableaux de lois (module `Distribution`) pour les différents modules. On fixe les lois en appelant la bonne fonction de création avec un fichier contenant ses paramètres. Ces tableaux seront utilisés lors de l'activation des modules. Voici un exemple de construction de tableaux et d'activation possible :

```
FILE * fic;
...

/* Creation du module Constant */
Constant_CreateConstEv (fic);

/* Creation des variables aleatoires */
DialerP [0] = Distribution_CreateBernouilli (fic);
DialerP [1] = Distribution_CreateExponential (fic);
DialerP [2] = Distribution_CreateExponential (fic);

OutcallP [0] = Distribution_CreateConstant(fic);

IncallP [0] = Distribution_CreatePoissonNonHomoCorrelated(fic);
IncallP [1] = Distribution_CreateExponential(fic);

ServersP [0] = Distribution_CreateGamma(fic);
ServersP [1] = Distribution_CreateUNURAN("/u/deslaura/maitrise/simulation/"
                                         "2elibrairie/logoutdure.dat");
ServersP [2] = Distribution_CreateConstant(fic);
ServersP [3] = Distribution_CreateConstant(fic);

/* Creation des objets des autres modules */
Dialer_CreateDialer (DialerP);
Waitqueue_CreateQueue();
Outcall_Create (OutcallP);
Incall_Create (IncallP);
Servers_CreateGroup (ServersP);
Statistic_CreateUnit ("statistiques journalieres");
```

Une fois que nos modules sont construits, nous pouvons faire la simulation de plusieurs jours. Au début d'une journée, on initialise les modules et on planifie l'ouverture du centre d'appels (`Constant_InitDay`), le début de la disponibilité des agents (`Constant_StartDay`),

le début de la composition d'appels sortants (`Constant_StartDialer`) et la fermeture du centre (`Constant_EndDay`). À l'ouverture du centre, on débute le processus des arrivées d'appels (`Incall_ArrivalScheduling`). Ensuite, on laisse les modules interagir entre eux jusqu'à la fermeture du centre. On recueille des mesures à la fin de chaque journée pour construire des estimateurs sur les mesures d'intérêt journalières. Il est possible d'imprimer un tableau informatif sur ces mesures en appelant les fonctions `Statistic_Report*`.

Nous avons construit un "makefile" permettant d'obtenir un exécutable du module `Call`. Cet exécutable prend en paramètre un nom de fichier d'entrée. Les informations importantes du fichier débutent sur la ligne suivant le signe `$$` et terminent avec une ligne vide. Voici un exemple de fichier de paramètres avec leur explication :

```
# Parametres de toutes les options possibles de simulation

# Nombre de repetition de la simulation
# et numéro de journée étudiée 0 = semaine, 1 = lundi,...
$$
10000 0

# Initday Startday StartDialer Endday
$$
7.91667 8.0 14.0 20.5

# Prob de rejoindre un client par intervalle + duree intervalle
$$
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0 30
0.27 30
0.27 30
0.28 30
0.29 30
0.29 30
0.3 30
0.33 30
0.37 30
0.4 30
```

```
0.38 30
0.41 30
0.41 30
0.41 30
```

```
# Moyenne (sec) du temps requis pour atteindre un client outbound ~ Exp
# + durée intervalle
$$
0.0 750
```

```
# Moyenne (sec) du temps requis avant de se rendre compte qu'un client outbound
# ne pourra pas etre rejoint ~ Exp + durée intervalle
$$
0.0 750
```

```
# Moyenne (sec) du temps d'attente acceptee du client sortant
# dans la file d'attente ~ Cte + durée intervalle
$$
5.0 750
```

```
# Parametres du processus d'arrivee Whitt voir memoire :
# parametres (alpha,beta) de W suivi de la fonction lambda(t)
# + durée intervalle
$$
```

```
28.9 0.0346
26.1 30
38.5 30
51.1 30
58.2 30
61.3 30
61.6 30
62.2 30
60.2 30
56.6 30
56.4 30
58.6 30
57.0 30
56.2 30
55.2 30
56.3 30
58.9 30
59.6 30
54.5 30
43.5 30
40.0 30
33.5 30
29.8 30
25.3 30
22.5 30
18.0 30
```

```
# Moyenne (sec) du temps d'attente acceptee du client entrant dans la
# file d'attente ~ Exp + durée intervalle
```

```
$$
```

```
400.0 60
400.0 30
700.0 60
600.0 150
500.0 390
100.0 30
50.0 30
```

```
# Paramètres du temps de service d'un appel inbound ~ Gamma + durée intervalle
```

```
$$
```

```
0.729 817.0 240
0.620 927.6 150
0.755 753.8 120
0.553 996.9 120
0.518 981.6 120
```

```
# Nombre d'agents disponibles par intervalle + durée intervalle
```

```
$$
```

```
11.3 30
18.1 30
23.0 30
25.9 30
26.5 30
27.0 30
26.4 30
26.3 30
23.9 30
24.2 30
30.6 30
29.4 30
25.7 30
24.0 30
24.2 30
24.7 30
22.2 30
18.7 30
17.0 30
9.4 30
4.1 30
3.4 30
2.9 30
3.1 30
3.0 30
```

Nombre d'agents disponibles par intervalle + durée intervalle

\$\$

0 30

0 30

0 30

0 30

0 30

0 30

0 30

0 30

0 30

0 30

0 30

0 30

6.1 30

12.6 30

17.3 30

20.1 30

18.4 30

16.6 30

12.8 30

19.0 30

19.5 30

18.5 30

17.3 30

19.5 30

18.0 30

###FIN