Université de Montréal

Simulation de centres de contacts

par
Eric Buist

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Rapport pour la partie orale
de l'examen pré-doctoral

Août, 2007

Université de Montréal
Faculté des études supérieures

Cet examen pré-doctoral intitulé:

Simulation de centres de contacts

présenté par:

Eric Buist

a été évalué par un jury composé des personnes suivantes:

Fabian Bastin,	président-rapporteur
Pierre L'Écuyer,	directeur de recherche
Pascal Vincent,	membre du jury

Examen accepté le: . . . . . .

TABLE DES MATIÈRES

CHAPITRE 1

INTRODUCTION

Un *centre de contacts* est un ensemble de ressources formant une interface entre un organisme et ses usagers. Plusieurs entreprises disposent d'un tel centre pour offrir des services à leurs clients tandis que des organismes gouvernementaux en possèdent pour les services de renseignements, d'urgence, etc. Les centres de contacts revêtent une grande importance économique, d'où le besoin de les analyser et d'en optimiser le rendement.

Avec l'accroissement de la complexité des systèmes, la simulation devient progressivement le seul outil capable de prendre tous les éléments en compte, mais les outils disponibles pour la simulation ne sont pas suffisamment performants pour effectuer des analyses et de l'optimisation efficacement. Pour simuler des centres de contacts plus facilement, nous avons alors, dans le cadre de notre projet de maîtrise, développé la bibliothèque *ContactCenters* qui permet de construire des simulateurs de centres de contacts dans le langage de programmation Java. En utilisant cette bibliothèque, nous avons également construit divers exemples de simulateurs dont un logiciel permettant de simuler, sans programmation Java, la plupart des centres d'appels que nous avons eu à traiter. ContactCenters est déjà plus rapide que tous les outils de simulation commerciaux que nous connaissons, mais son utilisation pose des difficultés aux gestionnaires et le logiciel n'est pas encore suffisamment performant pour effectuer de l'optimisation efficacement.

Dans le cadre de ce projet de doctorat, nous souhaitons améliorer ContactCenters sur quatre aspects : améliorer la conception et l'interface utilisateur de ContactCenters, mieux modéliser les centres de contacts pour une simulation plus réaliste, analyser la sensibilité des modèles aux changements de paramètres et améliorer l'efficacité des simulations.

Pour que ContactCenters intéresse un grand nombre de chercheurs et de gestion-

naires, le logiciel doit être très flexible et offrir une interface utilisateur conviviale. La flexibilité permet d'effectuer divers types d'expérimentations tandis que l'interface utilisateur sert à l'entrée des données, l'exécution des simulations et la sortie des résultats. Nous souhaitons pour cela développer un simulateur générique flexible, extensible avec peu de programmation et compatible avec les formats de données les plus courants.

Plusieurs aspects tels que l'absentéisme et la non-adhérence des agents, les recours des gestionnaires en cas de situation indésirable et les spécificités des télécopies, des courriers électroniques et d'autres types de contacts ne sont pas pris en compte dans notre modèle de simulation. Découvrir les détails importants qui ne sont pas actuellement modélisés est un grand défi en raison de la difficulté à obtenir des données et la complexité des centres de contacts.

Pour nous aider dans notre tâche de modélisation, nous allons analyser la sensibilité des principales mesures de performance aux changements de certains paramètres. Cela nous permettra par exemple de savoir dans quelles situations il est important de bien modéliser les temps de patience, les temps de service, le comportement individuel des agents, les détails de la politique de routage, etc. Si des aspects peuvent être négligés dans certains cas, cela nous permet de sauver du travail de modélisation et de simulation.

Nous souhaitons aussi augmenter l'efficacité de notre outil sans nécessairement négliger des aspects du modèle, en réduisant la variance ou le travail de simulation. Dans le premier cas, pour un temps de calcul identique ou légèrement supérieur, nous obtenons une variance beaucoup plus petite. Dans le second cas, nous obtenons une variance identique ou légèrement plus grande pour un temps de calcul significativement plus petit. Dans ce projet, nous allons en particulier tenter de combiner des techniques simples telles que des variables aléatoires communes et des variables de contrôle. Nous allons aussi expérimenter des techniques plus complexes telles que la stratification sur une fonction de plusieurs moyennes, les méthodes quasi-Monte Carlo et la scission (appelée *splitting* en Anglais). Nous allons aussi explorer des combinaisons plus complexes telles que la stratification avec les variables de contrôle.

Le reste de ce document est organisé de la façon suivante. Le chapitre suivant présente de façon plus détaillée le concept de centre de contacts ainsi que notre outil de simulation. Nous y présentons aussi de la notation qui sera utilisée par la suite dans le document. Le chapitre 3 résume nos objectifs d'amélioration de la conception du logiciel ContactCenters pour ce projet. Le chapitre 4 traite de nos objectifs reliés à la modélisation tandis que le chapitre 5 aborde l'analyse de sensibilité. Enfin, le chapitre 6 fait un survol des différentes techniques d'amélioration de l'efficacité que nous prévoyons exploiter tandis que le chapitre 7 explique comment nous pensons appliquer ces techniques à notre problème.

CHAPITRE 2

LES CENTRES DE CONTACTS

Un *centre de contacts* [7, 24, 44] est un ensemble de ressources telles que des lignes téléphoniques, des commutateurs, des routeurs, des employés et des ordinateurs servant d'interface de communication entre un organisme et ses usagers. La communication peut être effectuée via le téléphone, la télécopie, le courrier électronique, etc. Un centre de contacts ne traitant que des appels téléphoniques est appelé un *centre d'appels*.

De tels centres doivent traiter un grand nombre de requêtes de divers types, nécessitent une infrastructure technologique importante et emploient plusieurs préposés, d'où un coût de gestion élevé. D'un autre côté, la qualité du service offert affecte l'image de marque de l'organisme possédant un centre de contacts. Certains centres de contacts qui effectuent de la vente à distance peuvent même devenir une source de revenus pour une entreprise. L'importance économique des centres de contacts a déjà clairement été démontrée [21]. Les gestionnaires doivent donc établir un équilibre entre la réduction du coût et la qualité du service en effectuant des analyses de sensibilité et de l'optimisation.

Ce chapitre explique de façon plus détaillée en quoi consiste un centre de contacts et quels genres de problèmes ils posent. Nous y traitons également des types d'outils disponibles pour leur analyse pour ensuite nous concentrer sur notre solution, *ContactCenters*, que nous souhaitons améliorer dans le cadre de ce projet.

2.1 Concepts de base

Un *contact* consiste en une requête de communication entre un usager et un organisme. Les contacts *entrants* sont générés par des usagers tentant d'entrer en communication pour obtenir un service tel qu'une réservation ou du support technique. Les contacts *sortants* sont initiés de façon proactive par les employés ou, dans le cas des appels téléphoniques, par un système spécialisé appelé *composeur*. Ils permettent par

exemple la vente à distance ainsi que le rappel de clients. Les centres capables de traiter les deux types de contacts sont dits *mixtes*.

Afin de simplifier le traitement, chaque contact est classé selon un type représenté sous la forme d'un entier k entre 0 et $K-1$, où K est le nombre total de types de contacts supportés par un système particulier. Un contact est entrant si son type $k = 0, \ldots, K_{\mathrm{I}} - 1$ où $K_{\mathrm{I}} \leq K$ est le nombre de types de contacts entrants tandis qu'il est sortant si $k = K_{\mathrm{I}}, \ldots, K-1$. Le type de contact peut être déterminé en utilisant sa provenance (numéro de téléphone de l'appelant, site Web utilisé, etc.), les choix de l'usager dans des menus, etc. Il peut représenter la raison de la communication, l'importance du client, etc.

Le *service* d'un contact consiste en un traitement destiné à satisfaire la requête d'un usager. De nos jours, plusieurs requêtes peuvent être traitées entièrement par des systèmes automatisés, mais parfois, un usager peut manifester le besoin ou le désir de parler à un être humain. Dans ce contexte, le service comprend la phase de traitement automatique, le travail d'un employé pendant la communication avec le client et le travail que l'employé doit parfois effectuer après le dialogue.

Chaque employé, aussi appelé *agent*, fait partie d'un groupe $i \in \{0, \ldots, I-1\}$ définissant ses compétences. Il possède également certaines particularités qui peuvent affecter son efficacité et son horaire de travail. Au temps t de la journée, le groupe i contient $N_i(t)$ membres dont $N_{\mathrm{B},i}(t)$ sont en train de servir des contacts et $N_{\mathrm{I},i}(t)$ sont inoccupés. Il se peut que $N_i(t) < N_{\mathrm{B},i}(t) + N_{\mathrm{I},i}(t)$ si $N_{\mathrm{G},i}(t)$ agents terminent leur quart de travail (quittent le groupe) après avoir terminé le service en cours. Parmi les agents inoccupés, seuls $N_{\mathrm{F},i}(t) \leq N_{\mathrm{I},i}(t)$ sont connectés et disponibles pour de nouveaux services. Le nombre d'agents $N_i(t)$ est souvent plus petit que le nombre planifié en raison de retards, de pauses prolongées, etc. La figure 2.1 présente une vue schématique de ces différentes quantités. Nous pouvons également définir $N(t)$, $N_{\mathrm{G}}(t)$, $N_{\mathrm{B}}(t)$, $N_{\mathrm{I}}(t)$ et $N_{\mathrm{F}}(t)$ comme les équivalents des quantités précédentes pour tous les groupes d'agents du système.

Pendant leurs quarts de travail, les agents sont enregistrés auprès d'un *routeur* chargé d'acheminer les nouveaux contacts vers des agents libres et d'affecter des contacts en

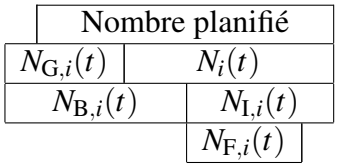

Figure 2.1 – Vue schématique des différentes quantités reliées à un groupe d'agents

attente à des agents devenus libres. Les règles de routage peuvent être très complexes, allant d'une simple liste de groupes d'agents spécifique à chaque type de contacts à une politique dynamique tenant compte de tout l'état du système pour prendre chaque décision. Évidemment, le routeur est une composante centrale dans un centre de contacts.

Un usager peut être servi par plusieurs agents avant d'obtenir satisfaction. Par exemple, un utilisateur éprouvant des problèmes techniques avec un logiciel pourrait parler à différents techniciens (simultanément ou séquentiellement) avant d'obtenir une solution. Un *retour* se produit lorsqu'un usager servi doit recontacter l'organisme pour obtenir un nouveau service ou tenter de nouveau de satisfaire sa requête initiale.

Dans le cas de communications différées comme les courriers électroniques, le service peut même être *préemptif*, c'est-à-dire qu'un agent peut interrompre une tâche consistant par exemple à répondre à un message pour se charger d'une tâche plus prioritaire comme traiter un appel téléphonique. Ainsi, en raison des retours et du service préemptif, les agents traitent parfois plusieurs usagers simultanément.

Un usager qui ne peut être servi immédiatement doit attendre en file. Il peut devenir impatient et décider d'abandonner, quittant le système sans recevoir de service. Dans ce cas, il peut tenter de recontacter le centre plus tard (on appelle cela *retrial* en Anglais) ou abandonner bel et bien, selon l'importance du service qu'il désire recevoir. Certains centres de contacts permettent également de laisser un message dans le but d'être recontacté ultérieurement. Cela forme des *files d'attente virtuelles* [53] qui sont traitées différemment des véritables files d'attente puisque le service des contacts dans de telles files est moins prioritaire que celui d'appels téléphoniques en attente. Dans le cas d'un appel téléphonique, un usager arrivé au moment où aucune ligne n'est disponible peut

également recevoir un signal occupé et être *bloqué* sans pouvoir attendre en file ou être servi.

Afin de simplifier la modélisation et l'estimation des paramètres des centres de contacts, l'horizon considéré (jour, semaine, mois, etc.) est habituellement divisé en périodes de quinze à soixante minutes pendant lesquelles les paramètres du processus d'arrivée, le nombre d'agents dans chaque groupe, les paramètres de la loi de probabilité pour les durées de service, etc. demeurent constants. Parfois, des statistiques sont également recueillies pour chacune des périodes séparément.

Plus précisément, le centre de contacts comprend P périodes dites *principales* représentant les heures d'ouverture. Chaque période principale $p = 1,\ldots,P$ correspond à l'intervalle de temps $[t_{p-1},t_p)$, où $t_0 < \cdots < t_P$. Dans le cas fréquent où chacune de ces périodes a une durée fixe d, $t_p = t_0 + pd$ pour $p = 1,\ldots,P$. Souvent, des contacts arrivent et se mettent en file avant l'ouverture du centre de contacts et le centre demeure actif après la fermeture pour traiter les services en cours et vider les files d'attente. C'est pourquoi nous définissons deux périodes additionnelles : la *période préliminaire* $[0,t_0)$ pendant laquelle le centre de contacts n'est pas encore ouvert et la *période de fermeture* $[t_P,T]$ pendant laquelle aucune arrivée ne se produit et les agents terminent leurs services.

2.2 Mesures de performance considérées

Les mesures de performance auxquelles s'intéressent les gestionnaires sont entre autres le niveau de service, le taux d'abandon, le temps de réponse moyen, le taux d'occupation des agents et le nombre d'équivalents à temps plein. Dans certains modèles, ils tentent également d'examiner la distribution des contacts entre les groupes d'agents. Toutes ces mesures sont définies sur un intervalle de temps constant $[t_1,t_2]$ correspondant à une demi-heure, une journée, un mois, etc.

Soit $S(t_1,t_2)$ le nombre de contacts servis dont le temps d'arrivée se situe dans cet in-

tervalle $[t_1,t_2]$ et soit $L(t_1,t_2)$ le nombre d'abandons pour les contacts arrivés pendant ce même intervalle. Ces deux quantités sont des variables aléatoires suivant une loi de probabilité inconnue en général. Pour analyser le comportement de telles variables, nous devons d'abord en recueillir un certain nombre d'observations sur lesquelles nous pouvons ensuite calculer des statistiques. En particulier, la moyenne de plusieurs observations de $S(t_1,t_2)$ permet d'estimer l'espérance notée $\mathbb{E}[S(t_1,t_2)]$ correspondant à la moyenne si nous disposions d'un nombre infini d'observations. Nous pouvons également estimer la variance de $S(t_1,t_2)$ qui est notée $\mathrm{Var}[S(t_1,t_2)]$.

Soit maintenant $S_{\mathrm{G}}(s,t_1,t_2)$ le nombre de contacts qui ont été servis après avoir attendu au plus s unités de temps. En particulier, $S_{\mathrm{G}}(0,t_1,t_2)$ correspond au nombre de contacts servis dès leur arrivée. Nous définissons également $S_{\mathrm{B}}(s,t_1,t_2) = S(t_1,t_2) - S_{\mathrm{G}}(s,t_1,t_2)$ comme étant le nombre de contacts servis après avoir attendu plus de s unités de temps. Contrairement à $S_{\mathrm{G}}(s,t_1,t_2)$ et $S_{\mathrm{B}}(s,t_1,t_2)$ qui sont des variables aléatoires, s est une constante correspondant à un seuil aussi appelé *temps d'attente acceptable*. De la même façon, nous pouvons définir $L_{\mathrm{G}}(s,t_1,t_2)$ comme le nombre d'abandons avant le seuil et $L_{\mathrm{B}}(s,t_1,t_2) = L(t_1,t_2) - L_{\mathrm{G}}(s,t_1,t_2)$.

Le *niveau de service* sur lequel Bell Canada et le CRTC [18] se sont entendus est

$$\frac{S_{\mathrm{G}}(s,t_1,t_2)}{S(t_1,t_2)+L_{\mathrm{B}}(s,t_1,t_2)}$$

où $[t_1,t_2]$ correspond à un mois. Pour analyser cette nouvelle variable aléatoire, nous devons encore une fois calculer des statistiques sur un certain nombre d'observations. Il est d'usage d'utiliser le rapport d'espérances

$$g_1(s) = \frac{\mathbb{E}[S_{\mathrm{G}}(s,t_1,t_2)]}{\mathbb{E}[S(t_1,t_2)+L_{\mathrm{B}}(s,t_1,t_2)]}, \tag{2.1}$$

car cela correspond au niveau de service à long terme, sur un nombre infini de mois. De plus, cette mesure ne dépend pas de la longueur de l'horizon si bien que nous pouvons approximer le niveau de service mensuel en simulant des journées indépendantes au lieu

de mois entiers.

D'autres définitions du niveau de service sont possibles, par exemple

$$g_2(s) = \frac{\mathbb{E}[S_{\mathrm{G}}(s,t_1,t_2) + L_{\mathrm{G}}(s,t_1,t_2)]}{\mathbb{E}[S(t_1,t_2) + L(t_1,t_2)]}. \tag{2.2}$$

Soit $A(t_1,t_2)$ le nombre d'arrivées pendant l'intervalle considéré. Si aucun contact n'est bloqué, $A(t_1,t_2) = S(t_1,t_2) + L(t_1,t_2)$. Le *taux d'abandon* à long terme est alors

$$\ell(t_1,t_2) = \frac{\mathbb{E}[L(t_1,t_2)]}{\mathbb{E}[A(t_1,t_2)]} \tag{2.3}$$

Une variante de ce taux consiste à remplacer $L(t_1,t_2)$ par $L_{\mathrm{B}}(s,t_1,t_2)$ pour omettre les abandons se produisant trop rapidement et qui peuvent être causés par des défaillances du système ou une quantité inévitable d'usagers impatients.

Le *temps de réponse moyen* à long terme est quant à lui défini comme

$$w(t_1,t_2) = \frac{\mathbb{E}[W_{\mathrm{S}}(t_1,t_2)]}{\mathbb{E}[S(t_1,t_2)]}, \tag{2.4}$$

où $W_{\mathrm{S}}(t_1,t_2)$ est la somme des temps d'attente pour tous les contacts servis arrivés pendant l'intervalle $[t_1,t_2]$. Toutes ces mesures peuvent être aussi définies pour un type de contact k particulier en ne comptant que les contacts de ce type plutôt que tous les contacts.

Le *taux d'occupation des agents* à long terme est quant à lui défini par

$$o(t_1,t_2) = \frac{\text{Nombre moyen d'agents occupés}}{\text{Nombre moyen d'agents connectés}} = \frac{\mathbb{E}\left[\int_{t_1}^{t_2} N_{\mathrm{B}}(t)\,dt\right]}{\mathbb{E}\left[\int_{t_1}^{t_2} (N_{\mathrm{B}}(t) + N_{\mathrm{F}}(t))dt\right]}. \tag{2.5}$$

De façon semblable, nous pouvons définir le nombre moyen d'agents membres d'un groupe comme

$$\bar{N}(t_1,t_2) = \frac{\mathbb{E}\left[\int_{t_1}^{t_2} N(t)\,dt\right]}{t_2 - t_1}. \tag{2.6}$$

Soit

$$\tilde{N}(h,t_1,t_2) = \bar{N}(t_1,t_2)\frac{t_2-t_1}{h} \tag{2.7}$$

le nombre d'*équivalents à temps plein* (*full-time equivalents* ou FTE en Anglais) si chaque agent a un quart de travail de durée h pendant l'intervalle $[t_1,t_2]$. Le taux d'occupation et le nombre d'agents moyens peuvent aussi être définis pour un groupe d'agents i particulier en ne comptant que les agents de ce groupe plutôt que tous les agents.

Soit maintenant $S_{k,i}(t_1,t_2)$ le nombre de contacts de type k servis par des agents du groupe i et soit $S_k(t_1,t_2)$ le nombre de contacts de type k servis par n'importe quel agent. Pour chaque paire (k,i), nous pouvons estimer

$$s_{k,i}(t_1,t_2) = \frac{\mathbb{E}[S_{k,i}(t_1,t_2)]}{\mathbb{E}[S_k(t_1,t_2)]}, \tag{2.8}$$

la fraction à long terme des contacts de type k servis par des agents du groupe i. Dans certains modèles où $K = I$, nous souhaitons favoriser le service des contacts de type k par des agents du groupe k, mais les contacts peuvent déborder vers d'autres groupes en cas de surcharge du groupe primaire. Dans ce contexte, les gestionnaires s'intéressent particulièrement à $s_{k,k}(t_1,t_2)$ qui est appelé *call match* en Anglais.

Du côté des contacts sortants, les gestionnaires s'intéressent surtout au taux de *mismatch* observé, c'est-à-dire la proportion d'usagers rejoints à un moment où aucun agent n'est disponible pour les servir par rapport au nombre total d'usagers rejoints. Ils examinent aussi le nombre d'usagers rejoints et l'impact des contacts sortants sur le niveau de service des contacts entrants.

2.3 Outils d'analyse

Les gestionnaires de centres de contacts désirent analyser de tels systèmes et en augmenter le rendement afin de réduire les coûts et augmenter la qualité du service. Pour cela, ils doivent considérer divers scénarios pour effectuer des analyses de sensibi-

lité. Ils sont également amenés à résoudre des problèmes d'optimisation, la plupart du temps sans disposer d'outils logiciels adéquats. Un problème courant consiste à trouver le nombre approprié d'agents pour que le niveau de service soit de 80% avec un temps d'attente acceptable de 20 secondes.

Avec les premiers centres ne traitant qu'un seul type d'appel téléphonique, des formules analytiques fondées sur la théorie des files d'attente étaient utilisées pour effectuer l'analyse sous des hypothèses simplificatrices fortes. Par exemple, le modèle Erlang C, couramment utilisé, considère que les arrivées suivent un processus de Poisson, que les temps de service sont indépendants et suivent la loi exponentielle et qu'aucun abandon n'est autorisé, ce qui est plutôt irréaliste. De plus, ces formules considèrent habituellement que le centre d'appels fonctionne dans des conditions identiques depuis un temps infini, ce qui ne prend pas en compte la non-stationnarité, c'est-à-dire la variation par rapport au temps des paramètres du centre. En réalité, les conditions changent régulièrement et nous souhaitons estimer la performance sur un horizon fini, par exemple une journée ou un mois. Les arrivées ne suivent pas toujours le processus de Poisson [5] et les temps de service ne sont pas toujours exponentiels. De plus, il est la plupart du temps trop coûteux de former tous les employés pour servir tous les contacts, même si cela pourrait augmenter la qualité de service en théorie [51]. Ainsi, bien que les centres de contacts sont encore modélisés par un système de files d'attente, seule la simulation peut fournir des évaluations précises tenant compte de toute la complexité. Malheureusement, faute d'outils adéquats, plusieurs gestionnaires de centres d'appels emploient encore les formules analytiques même lorsque leurs hypothèses ne sont pas vérifiées.

L'optimisation, quant à elle, demande d'évaluer la performance du centre de contacts plusieurs fois avec des paramètres différents. Ceci est possible avec des formules analytiques, mais si la simulation est mise en œuvre, il est nécessaire d'effectuer des milliers, voire des millions de réplications pour aboutir à un résultat précis. Il est alors important, pour y parvenir, de disposer d'outils très rapides. L'optimisation vise en premier lieu le nombre d'agents par intervalle de temps (aussi appelé *staffing* en Anglais [4]),

la construction d'horaires pour les agents [6], mais elle s'étend également au niveau du routage [33].

Pour donner une idée du temps pris par la simulation pendant l'optimisation, prenons par exemple la technique décrite dans [6]. Celle-ci résout un programme linéaire correspondant à une version simplifiée du problème d'affectation des quarts de travail aux agents qui tente de minimiser le coût tout en satisfaisant les contraintes de niveau de service. La technique utilise ensuite la simulation pour évaluer la réalisabilité de la solution et ajoute des coupes au programme linéaire simplifié en utilisant le sous-gradient du niveau de service par rapport au vecteur d'affectation des agents. Ce processus de coupe est répété jusqu'à obtenir une solution réalisable. Si cette méthode est appliquée sur un centre d'appels avec 20 types d'appels, 35 groupes d'agents et 52 périodes, avec seulement 300 réplications pour le test de réalisabilité et 20 réplications pour chaque composante de sous-gradient, il faut malgré tout $300 + 20 * 35 * 52 = 36\,700$ réplications par itération. Avec ces paramètres, l'optimisation a exigé cinq heures de temps de calcul.

Un centre de contacts peut bien entendu être modélisé grâce à un logiciel de simulation générique, mais cette tâche nécessite un énorme travail de conception et même de programmation. Il existe heureusement des logiciels spécialisés qui supportent la simulation de la plupart des centres de contacts d'aujourd'hui. Arena Contact Center Edition de Rockwell [47] et ccProphet de NovaSim [46] sont des exemples de tels logiciels. Toutefois, de nouveaux cas qui n'étaient pas prévus initialement peuvent survenir à n'importe quel moment et s'avérer difficiles à traiter sans recourir à des mécanismes de bas niveau qui dépendent du logiciel choisi et qui peuvent nécessiter la mise à jour vers une version plus complète (et plus coûteuse) du produit. Les logiciels commerciaux sont également formés d'un grand nombre de couches superposées, interconnectées et difficiles à séparer qui peuvent diminuer la performance.

2.4 Éléments de ContactCenters

De notre côté, nous avons développé la bibliothèque ContactCenters [11, 12, 13] an Java. Ce langage est puissant, largement utilisé et très bien supporté. Nous avons employé la bibliothèque SSJ [32, 35, 40] comme système de simulation en Java pour la génération des nombres aléatoires, la gestion de la liste d'événements et la collecte statistique. Grâce à l'héritage, les classes de ContactCenters peuvent facilement être étendues sans les récrire en entier. Un simulateur peut tirer parti de Java pour accéder à un grand nombre de bibliothèques d'optimisation, d'analyse statistique, ainsi qu'à des outils de construction d'interfaces graphiques. Grâce aux optimisations des récentes machines virtuelles Java, un simulateur écrit en Java s'exécute beaucoup plus rapidement qu'un modèle conçu grâce aux outils commerciaux les plus utilisés et fondés sur un langage complètement interprété et peu répandu.

ContactCenters est formée de composantes indépendantes qui sont reliées entre elles au moment de construire un programme simulant un modèle précis et détaillé de centre de contacts. Un tel programme peut également intégrer des techniques d'amélioration de l'efficacité. Ces composantes représentent les contacts (appels, télécopies, etc.), les processus d'arrivée, le composeur d'appels sortants, les groupes d'agents, files d'attente et le routeur. Chaque contact est représenté par une entité, c'est-à-dire un objet, avec son propre ensemble d'attributs prédéfinis que l'utilisateur peut étendre si nécessaire. Les sources de contacts (processus d'arrivée et composeur) construisent de tels contacts et les envoient au routeur qui se charge de mettre les contacts en service auprès d'agents ou les insérer dans des files d'attente pour les traiter plus tard. Le routeur peut quant à lui signaler les contacts sortants du centre à un système de collecte statistique.

Le programmeur peut facilement construire un observateur et l'enregistrer auprès de ces composantes pour, par exemple, connaître les contacts qui sont créés, ceux qui sortent du système, suivre l'état des agents, etc. En fait, tout le couplage entre les composantes du système est effectué à l'aide d'observateurs [23], ce qui permet un maximum de

flexibilité. De cette façon, les composantes peuvent être testées, améliorées et remplacées indépendamment des autres.

Par contre, écrire un programme Java pour simuler un centre de contacts peut être long et n'est pas à la portée de tous les gestionnaires. Un tel programme doit aussi répondre à un certain nombre de normes pour pouvoir interagir de façon générale avec d'autres programmes tels qu'un optimiseur. Il est alors important de disposer d'un simulateur précompilé le plus général possible permettant de traiter les cas les plus courants et d'interagir facilement avec d'autres outils. En utilisant les composantes de ContactCenters, nous avons construit un tel simulateur qui permet, en utilisant des fichiers de configuration dans le format XML [54], de traiter la plupart des centres d'appels courants.

Ce simulateur se restreint à un modèle particulier mais assez général de centre d'appels mixte supportant K_{I} types d'appels entrants et K_{O} types sortants, avec I groupes d'agents et P périodes principales de durée fixe. La plupart des paramètres sont spécifiés seulement pour les périodes principales. Pendant la période préliminaire, aucun agent n'est en service et pour les autres paramètres, les valeurs de la première période principale sont utilisées. Pendant la période de fermeture, les paramètres de la dernière période principale sont employés.

L'usager peut employer n'importe quelle loi de probabilité de SSJ pour les durées de patience et les durées de service, mais il doit choisir les processus d'arrivée, les politiques de composition d'appels sortants et la politique de routage parmi des listes de politiques prédéfinies. Il peut par contre paramétrer les différents processus avec des valeurs numériques.

Le simulateur calcule différentes statistiques comme le nombre d'appels produits, servis, bloqués ou ayant abandonné. Ces statistiques sont utilisées entre autres pour estimer les mesures de performance de la section 2.2. Chaque statistique est calculée pour chaque période p ainsi que pour tout l'horizon. Chaque événement relatif à un appel est compté dans la période de son arrivée et non dans celle où l'événement se produit.

Ceci est nécessaire pour éviter d'introduire un biais dans les estimateurs de rapports d'espérances. Par exemple, si plusieurs appels arrivaient pendant la période p et étaient servis pendant la période $p+1$, la valeur du niveau de service dans la période $p+1$ pourrait dépasser 100% si tous les événements étaient comptés dans la période où ils se produisent.

La simulation peut être effectuée de façon stationnaire pour une seule période de durée supposément infinie dans le modèle, en utilisant la méthode des moyennes par lots pour obtenir des intervalles de confiance, ou pour tout l'horizon, avec un nombre donné de réplications indépendantes. Dans le cas stationnaire (horizon infini), le simulateur est initialisé avec les paramètres pour une période et ces paramètres demeurent fixes tout au long de l'expérience. Dans le cas non stationnaire (horizon fini), les paramètres peuvent changer d'une période à l'autre. Simuler sur horizon infini ne semble pas naturel pour des centres de contacts, mais il peut s'avérer utile de le faire pour comparer les résultats de simulation avec des approximations considérant souvent le système comme stationnaire.

CHAPITRE 3

AMÉLIORATION DU LOGICIEL DE SIMULATION

Pour rendre ContactCenters accessible au plus grand nombre possible de chercheurs et de gestionnaires, nous devons faire face à deux ensembles de besoins distincts et parfois contradictoires. D'un côté, les chercheurs aimeraient disposer d'un outil très flexible leur permettant d'expérimenter de nouveaux modèles avec un minimum de travail de programmation tandis que de l'autre, les gestionnaires de centres de contacts ont besoin d'un outil très simple d'utilisation se fondant sur un modèle réaliste de centre de contacts, ne nécessitant aucune programmation et masquant même les aspects les plus complexes du processus de simulation. Malheureusement, la flexibilité peut facilement réduire la simplicité d'utilisation et les aspects complexes à masquer aux gestionnaires peuvent être nécessaires pour les chercheurs. Nous disposons déjà d'un outil flexible permettant d'expérimenter divers scénarios, mais plusieurs chercheurs éprouvent de la difficulté à écrire des programmes en Java. Nous espérons résoudre ces problèmes en ajoutant des couches logicielles au-dessus de la bibliothèque plutôt que la reconcevoir complètement. Nous aimerions en particulier faciliter l'implantation de nouvelles politiques de routage, accroître la flexibilité de notre simulateur générique précompilé et améliorer l'interopérabilité du même simulateur avec d'autres applications.

3.1 Politique de routage générique

Nous avons constaté suite à des requêtes des gestionnaires de Bell Canada que l'implantation de nouvelles politiques de routage est relativement long et les risques d'erreurs sont nombreux. Des aspects tels que les délais de débordement, l'attente dans plusieurs files et les files d'attente virtuelles, qui sont utilisés dans les routeurs de certains centres de contacts, sont actuellement difficiles à mettre en place avec ContactCenters. Nous disposons certes d'une politique relativement générale prenant plusieurs de ces aspects

en charge, mais il est difficile de l'étendre pour y ajouter de nouveaux aspects.

Pour le moment, ContactCenters offre deux niveaux de complexité pour les politiques de routage : utiliser une politique existante, prédéfinie, et la paramétrer selon ses besoins ou encore implanter sa propre politique sous la forme d'une classe Java, possiblement à partir d'une politique existante. Nous allons offrir un niveau intermédiaire en développant une politique de routage générique qui permettra à l'usager de construire des scripts de routage un peu comme sous Arena Contact Center Edition de Rockwell. De tels scripts de haut niveau seront formés par l'interconnexion de nœuds du genre « Mettre le contact en file auprès du groupe d'agents 1 », « Attendre dix secondes », etc. Le script de routage résultant sera beaucoup plus simple et concis qu'une politique de routage implantée directement avec ContactCenters.

Chaque nœud pourra recevoir de nouveaux contacts et disposera d'un certain nombre de sorties. Trois niveaux de complexité seront disponibles pour définir un nœud : utiliser un nœud prédéfini et le paramétrer, construire un nœud sous la forme d'un script interconnectant d'autres nœuds plus simples ou écrire une classe Java pour implanter le comportement du nœud. Il sera en effet relativement simple de construire de nouveaux nœuds pour, par exemple, définir des nouvelles politiques de distribution des contacts entre différentes files d'attente, acheminer des contacts de façon conditionnelle, etc.

Les nœuds prédéfinis et ajoutés par l'utilisateur pourront par la suite être interconnectés de façon statique, à l'aide d'une classe Java, ou de façon dynamique, en utilisant un fichier de configuration. En fait, le script de routage pourrait se trouver dans le même fichier XML qui contient les paramètres du modèle pour le simulateur générique (voir section 3.3). Pendant la conception de cette nouvelle politique de routage, nous garderons à l'esprit qu'une interface graphique pourrait être construite pour aider le gestionnaire à créer les scripts. L'implantation d'une telle interface sort du cadre de ce projet, mais nous pourrions superviser un étudiant dans cette tâche.

La difficulté principale pour concevoir cette politique est de définir les nœuds de base et les interconnexions possibles. Nous devons pour cela trouver un cadre englobant les

politiques de routage les plus courantes en nous inspirant de logiciels de simulation tels qu'Arena Contact Center Edition et de routeurs commerciaux. L'architecture choisie doit permettre de construire une politique de routage de façon intuitive, le plus souvent sans programmation Java, tout en ne réduisant pas la performance de façon trop importante.

3.2 Amélioration de la flexibilité du simulateur générique

Comme nous l'avons abordé à la section 2.4, nous avons utilisé les composantes de ContactCenters pour construire un simulateur générique qui peut être utilisé sans compiler de programme Java, simplement en écrivant des fichiers de configuration dans le format XML. Notre première version de ce simulateur était contenue dans une seule classe qui est vite devenue immense et difficile à gérer. Ce problème s'est manifesté pour la première fois lorsque nous avons tenté d'ajouter au simulateur des techniques de réduction de la variance. Au cours de l'année 2006, nous avons construit une deuxième version du simulateur formée de plusieurs classes. Il reste encore un important travail de documentation et de reconception afin de rendre ce simulateur extensible via un système de composantes enfichables, par exemple pour ajouter de nouvelles politiques de routage, de nouvelles statistiques, etc., sans recompiler le simulateur. Ceci permettrait entre autres d'expérimenter de nouvelles techniques tout en bénéficiant de toute l'infrastructure existante pour effectuer la simulation, produire les rapports statistiques, etc. Ajouter une fonctionnalité au simulateur générique est beaucoup plus simple et avantageux qu'écrire un tout nouveau programme pour traiter un cas particulier. Évidemment, certains cas complexes pourraient exiger la construction d'un tout nouveau programme, utilisant directement les composantes de ContactCenters.

Pour implanter des composantes enfichables, appelées *plug-ins* en Anglais, nous avons besoin d'interfaces ou de classes abstraites représentant chaque type de composante ainsi que d'un mécanisme pour construire les composantes à partir des paramètres stockés dans un fichier de configuration. Le mécanisme choisi doit permettre d'enregis-

trer de nouvelles composantes sans recompiler le simulateur et devrait idéalement fournir sur demande la liste des composantes enregistrées, par exemple pour permettre leur sélection dans une interface graphique. Par exemple, un générateur de nombres aléatoires, un objet capable de construire et configurer un routeur et un objet capable d'ajouter ou retirer une statistique précise du simulateur sont des types de composantes possibles. La réflexion, qui permet de se référer à des classes ou à des méthodes sans connaître leur nom lors de la compilation, est un exemple de mécanisme permettant de créer les composantes.

Plusieurs autres éléments peuvent encore être ajoutés au simulateur générique, notamment les files d'attente virtuelles, le regroupement des types de contacts et des groupes d'agents en segments pour afficher un rapport simplifié pour des centres de contacts complexes, le calcul de statistiques relatives à la fois aux types de contacts et aux groupes d'agents ainsi que des mécanismes simplifiant l'implantation de politiques de routage ou de composition d'appels sortants prenant en compte des statistiques cumulatives. Nous souhaitons également y incorporer les techniques de réduction de la variance (voir chapitres 6 et 7) s'appliquant dans un contexte général de façon à rendre ces techniques facilement utilisables. Nous ajouterons aussi des options pour les nouveaux aspects de modélisation liés à l'absentéisme et à la non-adhérence des agents ainsi qu'à la simulation des recours (voir chapitre 4). Chacune de ces extensions, prise individuellement, ne pose pas de grandes difficultés pour l'implantation, mais leur combinaison complique beaucoup le programme et pourrait en réduire sa performance.

Contrairement aux classes de base de ContactCenters qui sont faiblement couplées, le simulateur générique est formé de classes qui interagissent mutuellement. Ainsi, plus le simulateur grossit, plus il devient difficile à gérer et plus il devient important de trouver des façons de bien modulariser ses composantes. Il devient ainsi important et difficile de trouver la bonne architecture non pas seulement pour la bibliothèque ContactCenters mais aussi pour le simulateur précompilé qui l'utilise. C'est pourquoi la conception de ContactCenters est une contribution pour ce projet de doctorat.

Il peut ainsi devenir nécessaire de construire plusieurs simulateurs génériques adaptés à différentes tâches. Par exemple, nous pourrions développer un simulateur utilisant des compteurs au lieu de modéliser chaque contact comme un objet indépendant. Cela ne nous permettrait pas de recueillir directement les temps d'attente des contacts, qui sont utilisés pour estimer le niveau de service, mais le temps d'attente global peut être estimé à partir de la taille de la file par la loi de Little. Le simulateur permettrait aussi d'estimer la probabilité d'attente avec laquelle nous pourrions estimer le niveau de service. Un tel simulateur pourrait être beaucoup plus rapide que le simulateur générique actuel et pourrait remplacer avantageusement les formules d'approximation utilisées lors de l'optimisation.

Une autre avenue est de disposer d'un simulateur avec des composantes adaptées à différentes tâches. Nous avons déjà deux catégories de groupes d'agents, l'une traitant les agents individuels comme des objets et l'autre ne gérant que des compteurs indiquant le nombre d'agents occupés et libres. Notre logiciel définit aussi deux sortes de files d'attente, l'une implantée sous la forme d'une liste et l'autre sous la forme d'un ensemble trié pour les files à priorités multiples. Ce concept pourrait être appliqué à d'autres parties du système telles que le système de collecte statistique. En particulier, si nous simulons un modèle complexe sur un horizon très long divisé en périodes très courtes, par exemple un mois séparé en intervalles de cinq minutes, avoir un compteur statistique pour chaque type de contact et chaque période pourrait être trop coûteux en mémoire. Nous pourrions alors ajouter une version plus simple du collecteur statistique actuel qui recueillerait des observations pour des périodes plus longues et permettre à l'utilisateur de choisir entre les deux versions.

Nous pouvons également penser à appliquer la programmation orientée aspects [28] pour modulariser le simulateur. Ce nouveau paradigme introduit le concept d'*aspect* qui consiste en un ensemble de fragments de code appelés *advice* et se greffant à des endroits du programme appelés *points de jointure* (ou *join points* en Anglais). Un aspect sert à définir un comportement couvrant plusieurs endroits dans le programme, par exemple

la journalisation d'événements, le traitement d'erreurs, un aspect de modélisation tel que les recours qui nécessiteraient d'être pris en compte à plusieurs endroits pendant la simulation, etc. Nous pourrions mettre en œuvre la programmation orientée aspects en utilisant l'outil AspectJ [2] qui ajoute ce concept à Java.

3.3 Simplification de l'entrée des données

Le problème principal du simulateur générique réside sans aucun doute dans sa complexité d'utilisation. Écrire un fichier XML est certes plus simple que rédiger un programme Java, mais cela pose malgré tout des difficultés. Même avec un éditeur spécialisé qui permet d'éviter les erreurs de syntaxe et bon nombre d'erreurs grammaticales, les fichiers de paramètres demeurent malgré tout contre-intuitifs. De plus, le programme doit être démarré depuis la ligne de commande, une tâche à laquelle la plupart des utilisateurs ne sont plus accoutumés de nos jours.

L'idéal serait de disposer d'une interface graphique permettant à la fois de définir les paramètres du simulateur, d'exécuter des simulations et d'examiner les résultats. Cette interface pourrait aussi donner accès aux formules d'approximation les plus courantes, aux algorithmes d'optimisation existants, etc., qui ont déjà été implantés par un autre étudiant. Cette interface est en cours de développement par un stagiaire que nous supervisons durant l'été 2007.

La première étape pour simplifier l'entrée des paramètres consiste à construire un Schéma XML [49] pour contraindre la structure d'un fichier de paramètres. Un schéma indique quels éléments sont autorisés dans un fichier de paramètres, quels attributs sont permis pour chaque élément et quelles structures hiérarchiques peuvent être construites. Cela permet de valider les fichiers de paramètres de façon plus robuste et normalisée qu'un programme écrit manuellement pour traiter chaque paramètre séparément. Un schéma permet aussi à des éditeurs XML tels que <oXygen/> [43] et XMLSpy [1] de guider l'utilisateur dans la construction ou la modification d'un fichier de paramètres

et de valider eux-mêmes le fichier, sans démarrer pour cela le simulateur. Un schéma peut aussi contenir de la documentation pour chaque élément et attribut spécifié. Par contre, nos premières tentatives pour aboutir à un schéma nous ont mené à la nécessité de modifier le format actuel des fichiers de paramètres.

Le schéma choisi doit englober toutes les fonctionnalités actuelles du simulateur générique tout en laissant place à l'ajout de nouvelles fonctions. Idéalement, un fichier de configuration XML validé par ce schéma devrait permettre de démarrer le simulateur sans erreur, mais ceci est impossible en raison de contraintes impliquant plusieurs éléments qui ne peuvent pas être décrites par le schéma. Le format des fichiers spécifié par le schéma ne doit pas être trop verbeux puisque les fichiers pourront toujours être édités manuellement. Il est également crucial que le nouveau format soit ne change pas profondément par la suite puisque plusieurs composantes telles que le simulateur, les logiciels d'optimisation, une interface graphique pour éditer les paramètres, etc., en dépendront.

Nous avons développé un tel schéma durant l'été 2007 et avons adapté le simulateur pour l'utiliser. Pour cela, nous avons remplacé notre propre système de lecture de paramètres par *Java Architecture for XML Binding* (JAXB, [50]) version 2, un outil de liaison XML intégré à Java 6 mais aussi disponible en option sous Java 5. JAXB permet de transformer un document XML suivant un schéma déterminé en une hiérarchie d'objets et inversement de produire un document XML depuis des objets. La validation via un schéma XML peut être effectuée lors de ces deux opérations. Les classes utilisées par JAXB peuvent être écrites manuellement ou générées par le compilateur XJC à partir du schéma XML. Nous avons opté pour la seconde stratégie afin de simplifier l'ajout de paramètres ; il suffit alors de modifier le schéma et de redémarrer le compilateur XJC.

Nous avons ensuite dû écrire des méthodes de support pour convertir les objets les plus complexes représentant des tableaux bidimensionnels et des lois de probabilité vers des objets compatibles avec ContactCenters. Cette tâche accomplie, il nous a fallu adapter le simulateur pour utiliser la nouvelle hiérarchie d'objets de paramètres. Nous avons par la même occasion modifié son architecture en prévision des composantes enfichables

que nous avons abordées à la section précédente.

Nous avons également écrit des programmes pour convertir les fichiers de l'ancien format vers le nouveau, car nous disposions de certains gros fichiers dans l'ancien format qu'il aurait été long de convertir manuellement. Pour accomplir cette tâche, nous avons rédigé une feuille de style dans le langage XSL Transformations (XSLT, [15]) ainsi qu'un programme Java utilisant cette feuille de style pour la conversion. Le langage XSLT permet de définir des règles indiquant comment procéder pour générer un arbre XML résultant à partir d'un fichier XML source. Le langage XPath [16] est utilisé par XSLT pour déterminer l'ensemble des éléments et des attributs du fichier source sur lequel chaque règle du fichier XSLT s'applique ainsi que pour calculer des valeurs produites en sortie. Par contre, XSLT ne permettait pas de traiter les éléments les plus complexes de notre schéma si bien que nous avons été forcés de l'étendre par des fonctions d'extension. Une telle fonction est écrite dans un langage de programmation tel que Java et peut être appelée pendant l'application de règles pour produire du contenu.

Cette migration vers le nouveau format, qui est elle aussi une contribution pour ce projet de doctorat, est à peu près finie. Il nous reste des tests à effectuer ainsi que la documentation du simulateur à mettre à jour.

Le nouveau format de fichiers demeure malgré tout complexe et il arrive souvent que les gens n'utilisent qu'un sous-ensemble de ses possibilités. Dans ce cas, il serait envisageable de créer des formats plus simples mais plus limités et de convertir les fichiers vers le format du simulateur, en utilisant XSLT.

3.4 Importation de données depuis des sources diverses

Le simulateur doit pouvoir interagir avec d'autres applications couramment utilisées dans l'industrie tout en demeurant portable. En particulier, nous aimerions pouvoir importer des données depuis des sources externes telles des fichiers ou des bases de données. Bien que cela nous semble pour le moment irréaliste en raison de la diversité des

formats des données, l'idéal serait que le simulateur puisse recueillir lui-même les données dont il a besoin auprès des dispositifs chargés de la collecte des informations dans les centres d'appels.

Pour le moment, les paramètres du modèle doivent être entrés à la main en utilisant un éditeur de texte ou un éditeur XML. Ceci n'est pas très convivial et propice aux erreurs. L'interface graphique en cours de développement par un stagiaire durant l'été 2007 pourrait alléger significativement ce problème en autorisant le transfert d'informations depuis des logiciels tels que des tableurs, mais cela risque de ne pas suffire s'il est nécessaire de simuler plusieurs scénarios avec des paramètres différents.

Java permet sans difficulté de lire des fichiers textuels et dispose de bibliothèques intégrées pour accéder à des bases de données. Des bibliothèques existent également pour lire des fichiers au format Microsoft Excel. Nous pouvons ainsi écrire des programmes Java important des données de différentes sources sans perdre la portabilité. Par contre, ces programmes risquent de devenir complexes ou de devoir être personnalisés pour chaque utilisation.

Nous avons alors exploré la possibilité d'étendre le format XML de nos fichiers de paramètres pour permettre à l'utilisateur de remplacer les données par des instructions indiquant où se trouve l'information. Mais cela complique beaucoup le schéma XML, car le type de plusieurs éléments passe du simple tableau à une structure complexe. De même, lire les données depuis des sources externes, en particulier des bases de données, peut être long tandis qu'il n'est pas nécessaire de le faire à chaque simulation si les données ne changent qu'à des moments fixes. C'est pourquoi il nous semble plus judicieux de procéder comme suit. Au lieu de construire un fichier XML destiné directement au simulateur, l'utilisateur écrirait un gabarit contenant des paramètres et des instructions indiquant où recueillir les données manquantes. Il utiliserait ensuite un programme pour transformer ce gabarit en véritable fichier de paramètres. Évidemment, un tel gabarit pourrait être utilisé pour plusieurs expériences différentes, par exemple dans une application qui mettrait les donnés à jour quotidiennement à partir de bases de données. Le

langage XSLT (voir section 3.3), doté de quelques fonctions d'extension, pourrait servir à écrire de tels gabarits. Une autre possibilité consisterait à étendre les logiciels tels que des tableurs pour produire directement des fichiers XML exploitables par le simulateur.

3.5 Exportation des résultats vers des formats courants

À la fin de notre projet de maîtrise, en 2005, nous disposions d'un simulateur générique affichant les résultats sous une forme textuelle. Un programme Java pouvait aussi être rédigé pour appeler le simulateur et recueillir les résultats pour les analyser ou les formater de façon personnalisée. Ceci est toujours possible, mais analyser des résultats à partir de la sortie textuelle est fastidieux tandis qu'écrire un programme Java pour traiter les résultats n'est pas à la portée de touss les gestionnaires de centres de contacts. De même, il nous semblait utile de pouvoir exporter les résultats dans le format LaTeX pour pouvoir les intégrer à des documents destinés à ce système. Pour résoudre ces problèmes, le simulateur doit pouvoir exporter les résultats de façon à les rendre utilisables par des logiciels courants tels que Microsoft Excel et LaTeX, ce que nous avons déjà implanté au début de l'année 2007.

Il est également intéressant de pouvoir importer les résultats produits par le simulateur afin, par exemple, de permettre à une interface graphique de réafficher des résultats sauvegardés sans refaire la simulation. Cette importation est aussi cruciale pour permettre la rédaction de programmes automatisant la génération de tableaux synthétiques à partir de plusieurs scénarios. Par contre, analyser un fichier de sortie dans le format textuel ou Excel peut être difficile et propice aux erreurs. C'est pourquoi il est possible d'exporter les résultats vers un fichier XML réutilisable par un programme Java. Pour ce faire, un programme peut créer un objet de résultats à partir d'un fichier XML suivant le schéma approprié et en exploiter son contenu de la même façon qu'il accéderait aux résultats produits par un simulateur. Nous avons écrit le schéma pour ce nouveau type de fichier au cours de l'été 2007, mais il nous reste encore à adapter les classes actuelles

pour le prendre en charge.

CHAPITRE 4

AMÉLIORATION DE LA MODÉLISATION

Les gestionnaires chez Bell Canada ont remarqué que les niveaux de service et les taux d'occupation des agents obtenus par nos simulations ne correspondaient pas toujours à la réalité. Plus précisément, prenons un modèle définissant un seul type d'appel et un groupe d'agents, avec un processus d'arrivée de Poisson et des temps de patience et de service exponentiels.. Si nous utilisons les taux d'arrivées, les durées de service moyennes et le nombre d'agents pour chaque période correspondant à la réalité, nous obtenons la plupart du temps un niveau de service beaucoup trop élevé, par exemple 92% des appels servis après avoir attendu moins de 20 secondes au lieu du 80% escompté. Par contre, si nous essayons de réduire le nombre d'agents jusqu'à atteindre le niveau de service souhaité, nous obtenons un taux d'occupation plus élevé que celui observé dans la réalité, par exemple 93% au lieu de 80%. Cela trahit une modélisation incomplète du centre d'appels qui est due au manque de données et de certaines informations à notre disposition pour effectuer la modélisation.

Afin de corriger ce manque, nous devons trouver le ou les aspects importants à tenir compte dans le modèle en effectuant une analyse de sensibilité. Ceci semble principalement dû à l'absentéisme des agents, à leur non-adhérence et aux interventions manuelles des gestionnaires. D'autres facteurs tels que le processus d'arrivée des appels, la loi de probabilité des temps de service, etc., pourraient aussi entrer en jeu, Par exemple, dans le modèle précédent, nous avons tenté de remplacer le processus d'arrivée par un processus de Poisson doublement stochastique dont les taux suivaient une loi gamma. Avec ce processus appelé Poisson-gamma, il nous a fallu mettre 10% plus d'agents pour avoir 80% de niveau de service, mais nous avons obtenu un taux d'occupation de 80% au lieu de 93%. Pour que ContactCenters soit un outil utile au sein de l'industrie, il est crucial qu'il soit en mesure de traiter des systèmes correspondant à des centres d'appels réels.

Grâce à la collaboration des gens de Bell Canada dans le cadre d'une bourse à incidence industrielle du CRSNG, nous espérons rendre ContactCenters plus adéquat pour les gestionnaires. En particulier, nous pourrons adapter le logiciel en fonction de leur façon de voir les choses et faire en sorte que tous les aspects importants pour eux sont pris en compte pour simuler des modèles réalistes. Nous essaierons de modéliser et simuler divers centres d'appels de Bell Canada dans le cadre de ce projet. Nous espérons que cela nous permettra éventuellement de découvrir les aspects manquants pour obtenir des modèles réalistes. Gérer la complexité des centres d'appels réels est un défi important, car de multiples phénomènes dont l'impact n'est pas toujours quantifiable et significatif peuvent se produire. Par exemple, dans certains centres, le système de collecte de données ne comptabilise pas bien certains événements comme le temps passé par un agent à rappeler des clients. Dans d'autres, le nombre de types d'appels est très élevé et les agents d'un même groupe ne partagent pas nécessairement les mêmes aptitudes. Les routeurs utilisés sont produits par diverses entreprises et leur fonctionnement n'est pas connu de tous les gestionnaires. Par conséquent, aucun standard n'est utilisé pour décrire les politiques de routage, même de façon générale. Tous ces facteurs rendent difficile la collecte de données et même d'informations pour la modélisation. Nous tenterons malgré tout d'obtenir des informations et de les utiliser pour améliorer notre modèle. Nous espérons que nos contributions dans ce domaine inciteront les gestionnaires à rassembler davantage de données.

4.1 Absentéisme des agents

Jusqu'à présent, le nombre d'agents dans chaque groupe et chaque intervalle de temps était une constante donnée par l'utilisateur ou déterminée par un programme d'optimisation. En réalité, ce nombre peut varier de jour en jour puisqu'il peut arriver qu'un ou plusieurs agents ne se présentent pas à leur poste. Ce phénomène d'*absentéisme* fait en sorte que le nombre d'agents est une variable aléatoire qui pourrait changer d'une

réplication à l'autre d'une simulation.

Le modèle courant pour l'absentéisme est de réduire le nombre d'agents par un facteur fixé, ce qui permet en quelque sorte de réduire l'efficacité des agents ; il faut alors prévoir davantage d'agents pour obtenir une même qualité de service.

Pour modéliser l'absentéisme de façon plus réaliste, nous allons faire en sorte que chaque agent soit présent avec une certaine probabilité. Dans notre modèle le plus simple, le nombre d'agents de groupe i durant la période p suivra une loi binomiale. Nous tenterons ensuite de considérer la corrélation entre les périodes en faisant en sorte que le nombre d'agents du groupe i suive une loi multinomiale. Nous allons ensuite considérer un cas plus réaliste où nous disposons de l'horaire de travail des agents ; chaque agent sera alors présent avec une certaine probabilité, indépendamment des autres agents. Nous tenterons ensuite de modéliser la dépendance qu'il existe sans doute en réalité entre les agents.

4.2 Non-adhérence des agents

Un autre aspect important que nous ne modélisons pas encore bien est la *non-adhérence* des agents, c'est-à-dire que les agents présents ne suivent pas leurs horaires de travail à la lettre, par exemple en prenant des pauses imprévues. Ainsi, l'horaire de chaque agent est lui aussi stochastique.

Le modèle courant pour la non-adhérence est le même que pour l'absentéisme : réduire le nombre d'agents par un facteur fixé. Pour traiter la non-adhérence de façon plus réaliste, nous avons décidé en premier lieu d'examiner le comportement du modèle si les agents peuvent se débrancher pendant un certain temps après avoir servi des contacts. Dans ce modèle, si un agent termine un service, il peut se débrancher avec une certaine probabilité. Il reste alors inactif pendant une durée exponentielle. Nous avons découvert que ces déconnexions avaient un impact important sur le niveau de service et sur le taux d'occupation des agents en testant avec diverses probabilités et durées moyennes de dé-

connexion. Mais il fallait souvent fixer une probabilité ou une durée de débranchement trop élevées pour obtenir le niveau de service et le taux d'occupation correspondants à la réalité ; le modèle semblait irréaliste aux yeux des gestionnaires de Bell. Ceci semble dû au fait que notre modèle est trop simpliste. Par exemple, les paramètres de déconnexion pourraient varier dans le temps et même dépendre de l'état du système. En particulier, la probabilité d'une pause non planifiée pourrait augmenter en cas d'engorgement du système. La déconnexion pourrait aussi survenir sous d'autres conditions qu'une fin de service, par exemple si l'agent est en ligne depuis une durée aléatoire.

Nous aimerions trouver des modèles plus réalistes pour la non-adhérence. Si nous disposions de données adéquates, nous pourrions tester plus à fond le modèle précédent. Même en l'absence de données quantitatives, nous pouvons malgré tout obtenir des informations qualitatives et expérimenter à partir de cela. En particulier, nous avons appris que dans certains centres d'appels, chaque agent avait droit à un maximum de vingt minutes de non-adhérence. Une solution possible consisterait alors à considérer chaque agent individuellement et à retirer ces vingt minutes, réparties de façon aléatoire, de son temps de branchement.

Nous envisageons aussi de faire varier l'efficacité des agents en fonction de l'heure de la journée et d'autres paramètres du système. Par exemple, en fin de journées, les agents fatigués pourraient être plus lents à servir des contacts. Le temps de service pourrait aussi augmenter lorsque les agents deviennent trop occupés.

4.3 Simulation de recours

Les gestionnaires peuvent parfois intervenir pour faire face à des situations inacceptables, par exemple si le niveau de service diminue de façon importante ou si les agents ne sont pas suffisamment occupés. Ces situations peuvent être corrigées en modifiant les paramètres du composeur d'appels sortants ou du routeur, en appelant des agents additionnels, en demandant aux agents en ligne s'ils veulent prendre un congé pour la jour-

née, en modifiant la planification d'activités telles que des réunions, des formations, etc. En raison de ces recours, certains paramètres qui étaient constants jusque-là deviennent aléatoires et en corrélation avec d'autres facteurs tels que le nombre d'arrivées.

Par exemple, chez Bell Canada, nous avons travaillé sur le cas d'un centre d'appels comportant plusieurs groupes d'agents, chaque groupe disposant de sa propre file d'attente et correspondant à un fournisseur externe. Un appel arrivé était envoyé à un fournisseur avec une certaine probabilité et demeurait lié à ce dernier pour la durée de son séjour dans le centre. Les gestionnaires peuvent intervenir sur ces probabilités de façon manuelle afin d'équilibrer le niveau de service pour chaque fournisseur. On appelle cela *vendor service level management* (VSLM) en Anglais. Nous avons tenté d'élaborer une heuristique pour reproduire ces interventions manuelles, ce qui s'est avéré assez difficile.

Modéliser ces interventions manuelles est difficile, car les gestionnaires utilisent plusieurs facteurs qui ne sont pas tous considérés lors de la simulation et n'appliquent pas toujours une méthode rigoureuse pour prendre leurs décisions. Parfois, modéliser ces interventions apporte une contribution négligeable à la précision, surtout si l'horizon de simulation se restreint à une journée typique, mais cela peut devenir important surtout si nous modélisons les événements imprévus comme les pannes ou si nous simulons sur un horizon plus long. Découvrir la nature de ces interventions manuelles et déterminer quelles interventions sont importantes à simuler est une direction de recherche prometteuse dans laquelle nous pourrions apporter des contributions.

4.4 Gestion d'autres types de contacts

ContactCenters doit aussi pouvoir prendre en charge d'autres types de communications que les appels téléphoniques, par exemple les télécopies et la messagerie instantanée. À priori, rien ne semble empêcher un programmeur de construire un simulateur traitant ce type de contacts.

La difficulté réside dans la façon de modéliser un tel centre de contacts : comment faut-il effectuer le routage, quelle est la limite sur le nombre de communications qu'un agent peut traiter, un agent peut-il traiter un appel téléphonique et de la messagerie instantanée en même temps, etc. ? La construction de modèles pour les centres de contacts prenant plusieurs types de communication en charge semble nouvelle et laisse donc place à l'innovation. Nous allons construire des exemples de simulateurs pour modéliser ce genre de centres, en utilisant toutes les données que nous pourrons obtenir à leur sujet. Cela pourrait soulever des difficultés qui nous permettraient ensuite de décider s'il est nécessaire d'ajouter de nouvelles fonctionnalités à ContactCenters pour faciliter l'implantation.

CHAPITRE 5

ANALYSE DE SENSIBILITÉ

Comme nous l'avons abordé dans le chapitre précédent, le taux d'occupation des agents obtenu par nos simulations était souvent trop élevé par rapport à celui de la réalité. Dans notre modèle, les agents sont donc trop efficaces si bien que nous avons proposé des façons de réduire leur efficacité, directement par l'absentéisme et la non-adhérence ou indirectement en induisant de la variabilité dans le modèle.

Puisque rassembler des données pour construire un modèle est long et coûteux et que chaque détail inclus dans le modèle réduit la vitesse de simulation, il est judicieux de ne modéliser que ce qui est nécessaire pour atteindre notre objectif. Dans cette optique, nous souhaitons effectuer une analyse de sensibilité de modèles dans diverses conditions pour déterminer les aspects qui sont importants et ceux qui peuvent être négligés. Une telle analyse consiste à faire varier certains paramètres d'un modèle et d'observer le comportement des résultats en fonction de cette variation. Ceci nous est possible, car nous disposons d'un simulateur très flexible avec lequel nous pouvons faire plusieurs expérimentations. La difficulté de cette partie du projet ne consiste donc pas à obtenir des résultats mais plutôt à les interpréter et à décider sur quelles variables nous allons agir pendant les expériences.

Nous souhaitons en premier lieu déterminer les façons les plus efficaces d'augmenter la variabilité dans le but de réduire l'efficacité des agents simulés. Nous voulons aussi savoir s'il est nécessaire de connaître avec précision les lois de probabilité pour les temps de patience et les temps de service des contacts. Nous allons également examiner l'impact de certaines simplifications du modèle de simulation comme l'utilisation de compteurs au lieu d'objets, l'agrégation de types de contacts et de groupes d'agents, etc.

Comme nous allons le découvrir dans les prochaines sections de ce chapitre, il existe un nombre infini d'expérimentations que nous pourrions effectuer. Nous allons donc

nous limiter à des modèles simples pour lesquels nous pourrons interpréter les résultats. Nous espérons que nos découvertes s'appliqueront aussi à des modèles plus complexes.

Dans ce chapitre, nous allons expliquer comment nous prévoyons effectuer cette analyse de sensibilité, sur quels genres de modèles et pour quelles mesures de performance. Nous allons présenter un seul exemple de modèle, mais nous prévoyons expérimenter sur différents modèles inspirés d'exercices de simulation faits chez Bell Canada dans le cadre de ce projet. En particulier, nous aimerions analyser des exemples avec plusieurs types de contacts qui ne peuvent pas être servis par tous les agents.

5.1 Méthodologie

Soit $\theta \in \mathbb{R}$ un paramètre que nous souhaitons faire varier et soit $\nu(\theta)$ (une constante) la mesure de performance qui nous intéresse, en fonction de θ. Comme dans le chapitre suivant, $\nu(\theta)$ peut être une fonction de plusieurs espérances. Nous simulons le système n fois, indépendamment, pour calculer un estimateur $\hat{\nu}_n(\theta)$ (une variable aléatoire) de $\nu(\theta)$ pour différentes valeurs de θ. Par exemple, θ pourrait correspondre au temps de service moyen des contacts et $\nu(\theta)$, au niveau de service avec ce paramètre θ.

Nous allons perturber θ en lui ajoutant une constante $\delta \in \mathbb{R}$ et estimer la différence

$$\Delta = \nu(\theta + \delta) - \nu(\theta)$$

avec l'estimateur

$$\hat{\Delta}_n = \hat{\nu}_n(\theta + \delta) - \hat{\nu}_n(\theta).$$

La constante δ peut dépendre de θ. Nous pouvons aussi nous intéresser à la différence relative

$$\Delta_{\mathrm{REL}} = \frac{\nu(\theta + \delta) - \nu(\theta)}{\nu(\theta)}.$$

Nous utiliserons les variables aléatoires communes (voir section 6.2) pour réduire la variance sur la différence estimée. Dans ce cas, pour que les deux systèmes comparés

ne diffèrent pas trop, il faut que δ soit près de 0. Toute autre technique que nous allons développer pour calculer des sous-gradients dans le but de faire de l'optimisation (voir en particulier la section 6.8.2) pourrait être réutilisée pour notre analyse de sensibilité.

Nous prévoyons tester si perturber θ affecte significativement la mesure de performance et, dans certains cas, trouver une perturbation affectant la performance jusqu'à un certain point. Nous pourrions pour cela appliquer des tests d'hypothèses, mais ces derniers ne nous fourniraient que des réponses qualitatives : soit la différence testée serait significative, soit le test ne permettrait pas de décider. Pour avoir une idée de l'importance de la différence Δ, nous calculerons plutôt un intervalle de confiance $[I_1, I_2]$ de niveau $1-\alpha$ sur Δ pour l'estimateur $\hat{\Delta}_n$. Ici, I_1 et I_2 sont des variables aléatoires et nous allons supposer que $\mathbb{P}[I_1 \leq \Delta \leq I_2] \approx 1-\alpha$. Si $I_1 \leq \tau \leq I_2$, nous pouvons alors considérer que $\Delta = \tau$ avec un niveau de confiance de $1-\alpha$. Par exemple, si nous voulions savoir si, en multipliant le temps de service moyen par un certain facteur κ, le niveau de service se trouve perturbé de 5% avec probabilité 95%, nous fixerions τ et α à 5%.

Pour notre analyse de sensibilité, nous allons nous concentrer sur les mesures de performance souvent considérées par les gestionnaires de Bell Canada : le niveau de service $g_1(s,t_1,t_2)$, le taux d'abandon $\ell(t_1,t_2)$, le temps de réponse moyen $w(t_1,t_2)$ et le taux d'occupation des agents $o(t_1,t_2)$. Dans le cas d'un modèle où $K > 1$ et $I > 1$, nous pouvons aussi examiner $s_{k,i}(t_1,t_2)$, le nombre de contacts de type k servis par des agents du groupe i. En général, les gestionnaires souhaitent équilibrer les performances entre types de contacts et groupes d'agents. Dans certains cas, ils veulent aussi équilibrer la performance à l'intérieur d'intervalles de temps de l'horizon, par exemple dans chaque demi-heure de la journée. Si $K > 1$, nous pouvons donc examiner le niveau de service, le taux d'abandon et le temps de réponse moyen pour chaque type de contact et pour $I > 1$, nous pouvons étudier le taux d'occupation et le nombre d'équivalents à temps plein pour chaque groupe d'agents. Ces mesures ont été définies à la section 2.2.

Le comportement du système pourrait varier en fonction de la charge de travail des agents qui peut être mesurée par leur taux d'occupation. Nous nous attendons à ce que

la sensibilité des mesures de qualité de service soit plus faible dans les cas extrêmes où le trafic est très bas ou très élevé et allons vérifier cela empiriquement. C'est pourquoi nous allons faire varier les taux d'arrivée pendant les tests.

Parfois, les gestionnaires s'intéressent aussi à la sensibilité du nombre d'équivalents à temps plein nécessaires pour obtenir un niveau de service fixé, par exemple 80% en 20 secondes. Dans ce contexte, nous pouvons tenter d'ajuster les agents pour avoir le bon niveau de service après chaque changement de paramètre et examiner le nombre d'agents requis en plus des autres mesures de performance. Mais ceci ne peut être testé efficacement qu'en utilisant des programmes d'optimisation rapides dont le développement sort du cadre de ce projet de doctorat. Par contre, un autre étudiant est en train de réaliser un projet de doctorat sur ce sujet si bien que nous pourrions utiliser son travail pour notre analyse. Même sans logiciel d'optimisation, nous pouvons tenter de trouver le bon nombre d'agents par essai et erreur pour des modèles simples.

5.2 Analyse d'un modèle avec un type de contact et un groupe d'agents

Cet exemple, tiré du guide d'utilisation de SSJ [32] et utilisé dans [36], définit un seul type de contact, un seul groupe d'agents mais $P = 13$ périodes principales d'une heure. Les contacts arrivent selon un processus de Poisson à un taux aléatoire $B\lambda_p$ durant la période p. Les taux de base λ_p sont déterministes tandis que B est une variable aléatoire dont la loi de probabilité est gamma avec paramètres (α_0, α_0). Générée au début de chaque jour, la variable B a une moyenne de 1, une variance de $1/\alpha_0$ et représente le facteur d'achalandage du système [5]. Si $B > 1$, le taux d'arrivée des contacts est plus élevé que d'habitude. Si $B < 1$, il est inférieur à la moyenne. Ce processus permet d'augmenter la variance du nombre d'arrivées A puisque

$$\mathrm{Var}[A] = \mathrm{Var}[\mathbb{E}[A \mid B]] + \mathbb{E}[\mathrm{Var}[A \mid B]] = a^2\mathrm{Var}[B] + a,$$

où $a = \mathbb{E}[A]$.

Les contacts ne pouvant être servis immédiatement sont mis dans une file de type FIFO (*First in First Out* ou premier arrivé, premier servi) et abandonnent après un certain temps de patience. Les temps de patience sont i.i.d. et sont générés de la façon suivante : avec probabilité ρ, le temps de patience est 0, c'est-à-dire que le contact concerné abandonne immédiatement s'il ne peut pas être servi dès son arrivée. Avec probabilité $1-\rho$, le temps de patience est exponentiel de moyenne $1/\nu$. Les temps de service sont exponentiels avec moyenne $1/\mu$.

Pendant la période p, N_p agents sont disponibles pour servir des contacts. Si, à la fin de la période p, le nombre d'agents occupés est plus grand que N_{p+1}, les services en cours sont complétés et de nouveaux contacts ne sont acceptés que lorsque le nombre d'agents occupés devient inférieur à N_{p+1}. Pendant la période préliminaire, aucun agent n'est disponible tandis que $N_{P+1} = N_P$. Le tableau 5.I présente les paramètres avec lesquels nous prévoyons tester cet exemple. Avec ces paramètres, le nombre total espéré d'arrivées est 1 660.

Changement des lois de probabilité. Si les durées de service sont plus courtes, le taux d'occupation des agents diminue, mais le niveau de service augmente. Si nous souhaitons faire diminuer le taux d'occupation tout en gardant le même niveau de service, nous pouvons tenter d'utiliser une loi de probabilité autre d'exponentielle pour les temps de service afin d'en augmenter la variabilité. Par exemple, la loi gamma généralise la loi

Tableau 5.I – Paramètres de l'exemple avec un type de contact et un groupe d'agents

p	0	1	2	3	4	5	6	7	8	9	10	11	12
N_p	4	6	8	8	8	7	8	8	6	6	4	4	4
λ_p	100/h	150/h	150/h	180/h	200/h	150/h	150/h	150/h	120/h	100/h	80/h	70/h	60/h

Périodes	Treize périodes d'une heure
α_0 pour B	10
Probabilité ρ d'abandon immédiat	0,1
Temps de patience moyen	16min40s
Temps de service moyen	1min40s

exponentielle et permet de fixer la variance en plus de la moyenne.

Si les durées de patience sont réduites, le nombre d'abandons augmente, ce qui libère des agents tout en diminuant le niveau de service. Nous pouvons aussi essayer de changer la loi de probabilité des temps de patience afin d'en augmenter la variance comme avec les temps de service. Évidemment, la sensibilité au temps de patience dépend beaucoup du taux d'abandon dans le modèle. Si ce taux est petit, les changements auront peu d'impact.

Le processus d'arrivée est bien entendu une importante source de variabilité dans le modèle si bien qu'il vaut la peine d'essayer de le perturber lui aussi. Faire varier la variance du facteur d'achalandage est déjà une façon de changer le processus d'arrivée. Nous pourrions aussi tester avec des processus totalement différents comme le processus de Poisson sans facteur d'occupation B, le processus Poisson-gamma pour lequel le taux d'arrivée dans chaque période est une variable aléatoire de loi gamma, etc.

Paramètres des agents. Évidemment, tout changement dans les paramètres des agents peut affecter leur taux d'occupation si bien que nous allons tenter de faire varier ces paramètres. Nous pouvons d'abord faire varier le nombre d'agents en multipliant N_p par un facteur ε et en arrondissant le résultat. Cela permettrait d'évaluer la performance en fonction du nombre d'équivalents à temps plein. Ceci a pour but de déterminer à quel point il faut changer le nombre d'agents pour obtenir la performance souhaitée.

Nous allons également essayer de faire varier la probabilité de déconnexion des agents après chaque service ainsi que la durée moyenne de la déconnexion pour tester plus à fond notre modèle de la non-adhérence développé à la section 4.2.

Nous allons finalement comparer le comportement du système où un groupe d'agents est implanté avec des compteurs avec un système équivalent dans lequel un groupe d'agents représente chaque agent comme un objet individuel. La différence apparaît lorsque $N_i(t) < N_{\mathrm{B},i}(t)$ dans un groupe i et que tous les agents terminent leurs services en cours avant de quitter. Dans un groupe implanté avec des compteurs, aucun nouveau

contact n'est accepté tant que $N_i(t) < N_{\mathrm{B},i}(t)$. Par contre, si chaque agent est un objet indépendant, certains agents sont marqués pour quitter le système à la fin de leurs services et il se peut qu'un agent non marqué termine son service avant un agent marqué. Dans ce cas, des nouveaux contacts pourraient être acceptés même si $N_i(t) < N_{\mathrm{B},i}(t)$.

5.3 Un seul type de contact, plusieurs groupes d'agents

Nous allons tester des modèles définissant un type de contact mais plusieurs groupes d'agents. Bien entendu, si aucune distinction n'est faite entre les agents, par exemple en fixant un temps de service différent selon le groupe d'agents ou en ajustant la politique de routage, cela équivaut à n'avoir qu'un groupe d'agents. Nous allons nous limiter ici à des politiques couramment utilisées, car le nombre de politiques possibles est beaucoup trop grand pour que nous puissions toutes les tester.

La première règle de routage qui nous vient à l'esprit consiste à choisir l'agent avec le plus long temps d'inactivité, ce qui n'établit aucune distinction entre les agents si les durées de service ne dépendent pas d'eux. Nous devrions alors observer un taux d'occupation équilibré pour les agents. Mais il se peut qu'en réalité, les taux ne soient pas équilibrés, d'où une différence du taux global. Nous pouvons alors tenter de reproduire cela en faisant varier les durées de service en fonction du groupe d'agents.

Une autre façon d'établir la distinction est d'employer une politique avec débordements : choisir un agent dans un premier groupe puis opter pour un agent dans un second si tous les agents du premier sont occupés et ainsi de suite. Nous allons considérer que l'ordre de consultation des agents est fixe. Pour diminuer le taux d'occupation des agents d'un groupe, nous pouvons tenter de réduire le nombre de contacts envoyés vers ce groupe en imposant un délai minimal d'attente aux contacts. Plus le délai de débordement pour un groupe i est élevé, plus les contacts doivent attendre longtemps avant de pouvoir être servis par les agents de ce groupe. Le taux d'occupation des agents du groupe i devrait alors diminuer tandis que celui des agents des groupes précédant i dans

la liste de débordement devrait augmenter.

5.4 Plusieurs types de contacts

Pour un maximum d'efficacité, il vaut mieux que tous les agents soient formés pour servir tous les contacts. En réalité, un agent ne peut servir qu'un nombre restreint de types de contacts. Cela peut réduire son taux d'occupation étant donné que sa charge de travail se voit ainsi réduite. Pour modéliser un centre de contacts de façon réaliste, il semble donc important de fixer des paramètres appropriés pour chaque type de contact et chaque groupe d'agents.

Par contre, plus il y a de types de contacts et de groupes d'agents, plus le système devient complexe. Nous nous demandons alors s'il est possible d'agréger certains types de contacts et groupes d'agents pour sauver du travail de modélisation. Pour examiner cela, nous allons comparer certains modèles complexes inspirés de centres d'appels chez Bell Canada avec des versions simplifiées agrégeant certains éléments. Nous espérons montrer empiriquement que dans certains cas, agréger est avantageux.

CHAPITRE 6

TECHNIQUES D'AMÉLIORATION DE L'EFFICACITÉ

Il existe un très grand nombre de techniques pour réduire la variance [10, 22, 25, 30, 34] dans les simulations. Par contre, la plupart de ces méthodes doivent être personnalisées pour les adapter à l'application donnée et il est parfois difficile de les utiliser pour un système complexe. Une autre avenue consiste à combiner plusieurs techniques, ce qui fait souvent surgir des problèmes insoupçonnés. De telles combinaisons ont été étudiées dans [8, 9] pour certains cas particuliers.

Dans ce chapitre, nous allons présenter diverses techniques permettant d'améliorer l'efficacité en réduisant la variance de façon générale. Les techniques présentées ici sont relativement connues et sont décrites dans [34] et les références bibliographiques qui s'y trouvent. Nous présentons ici les techniques dans un contexte général avec quelques exemples relatifs aux centres de contacts. Dans le chapitre suivant, nous exposerons plus en détails comment nous souhaitons tenter d'exploiter ces techniques pour la simulation de centres de contacts.

La plupart de ces techniques visent à réduire la variance asymptotique d'une fonction de plusieurs moyennes. Soit pour cela $\mathbf{X} = (X_1, \ldots, X_d) = (X_i)_{i=1}^d$ un vecteur aléatoire obtenu par simulation. Ce vecteur peut par exemple contenir le nombre de contacts servis, le nombre d'abandons, etc. Nous nous intéressons à l'espérance de ce vecteur, notée $\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}]$. Soit maintenant $g : \mathbb{R}^d \to \mathbb{R}$ une fonction continue et dérivable dans le voisinage de $\boldsymbol{\mu}$. La constante $\nu = g(\boldsymbol{\mu})$ est estimée par $\hat{\nu}_n = g(\hat{\boldsymbol{\mu}}_n)$, où $\hat{\boldsymbol{\mu}}_n$ est un estimateur asymptotiquement sans biais de $\boldsymbol{\mu}$, c'est-à-dire que $\mathbb{E}[\hat{\boldsymbol{\mu}}_n] \to \boldsymbol{\mu}$ si $n \to \infty$. Nous allons également supposer que $\hat{\boldsymbol{\mu}}_n \Rightarrow \boldsymbol{\mu}$, où $\Rightarrow$ dénote la convergence en loi de probabilité. Cela signifie que quand $n \to \infty$, la répartition de $\hat{\boldsymbol{\mu}}_n$ tend vers une loi discrète générant $\boldsymbol{\mu}$ avec probabilité 1.

En général, $\hat{\nu}_n$ est un estimateur biaisé de ν, à moins que $g(\boldsymbol{\mu})$ soit une fonction

linéaire. Un cas important de fonction linéaire est $g(\boldsymbol{\mu}) = \mu_i$, c'est-à-dire une composante individuelle de $\boldsymbol{\mu}$.

L'expression de $\mathrm{Var}[g(\mathbf{X})]$ est habituellement inconnue, mais nous pouvons obtenir une expression relativement simple pour la variance *asymptotique* de $\hat{v}_n$, c'est-à-dire la variance quand $n \to \infty$. Nous allons utiliser cette variance asymptotique comme approximation de la variance en supposant que n est grand. Pour cela, le théorème de Taylor [48] indique que si $\mathbf{X}$ est près de $\boldsymbol{\mu}$,

$$g(\mathbf{X}) = g(\boldsymbol{\mu}) + (\nabla g(\boldsymbol{\xi}))^{\mathrm{t}}(\mathbf{X} - \boldsymbol{\mu}) \approx g(\boldsymbol{\mu}) + (\nabla g(\boldsymbol{\mu}))^{\mathrm{t}}(\mathbf{X} - \boldsymbol{\mu}) \tag{6.1}$$

où $\boldsymbol{\xi} \in \mathbb{R}^d$ se trouve sur le segment reliant $\boldsymbol{\mu}$ et $\mathbf{X}$ et $\nabla g(\boldsymbol{\xi})$ est le gradient de $g(\boldsymbol{\xi})$ évalué à $\boldsymbol{\xi}$. Si nous supposons que le gradient est continu dans le voisinage de $\boldsymbol{\mu}$ et que la fonction est évaluée à $\hat{\boldsymbol{\mu}}_n$ avec $\hat{\boldsymbol{\mu}}_n \Rightarrow \boldsymbol{\mu}$ quand $n \to \infty$, alors $\boldsymbol{\xi} \Rightarrow \boldsymbol{\mu}$ si bien que $\nabla g(\boldsymbol{\xi}) \Rightarrow \nabla g(\boldsymbol{\mu})$. Nous pouvons alors supposer que l'approximation (6.1) devient exacte à la limite, ce qui nous donne une expression pour la variance.

$$\begin{aligned}
\sigma^2 &\stackrel{\text{def}}{=} \lim_{n\to\infty} n\mathrm{Var}[g(\hat{\boldsymbol{\mu}}_n)] = \lim_{n\to\infty} n\mathrm{Var}[g(\boldsymbol{\mu}) + (\nabla g(\boldsymbol{\mu}))^{\mathrm{t}}(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu})] \\
&= \lim_{n\to\infty} n\mathrm{Var}[(\nabla g(\boldsymbol{\mu}))^{\mathrm{t}}\hat{\boldsymbol{\mu}}_n] = \lim_{n\to\infty} n(\nabla g(\boldsymbol{\mu}))^{\mathrm{t}}\mathrm{Cov}[\hat{\boldsymbol{\mu}}_n]\nabla g(\boldsymbol{\mu}) \\
&= (\nabla g(\boldsymbol{\mu}))^{\mathrm{t}}\boldsymbol{\Sigma}_{\mathrm{XX}}\nabla g(\boldsymbol{\mu})
\end{aligned}$$

où $\boldsymbol{\Sigma}_{\mathrm{XX}} = \mathrm{Cov}[\mathbf{X}] = n\mathrm{Cov}[\hat{\boldsymbol{\mu}}_n]$ est une matrice de covariances de $d \times d$ définie positive. De plus, $\sigma^2 \approx \mathrm{Var}[g(\mathbf{X})]$ si $\mathbf{X}$ est près de $\boldsymbol{\mu}$, ce qui est vérifié avec $\hat{\boldsymbol{\mu}}_n$ quand n est grand. Si $g(\boldsymbol{\mu})$ est linéaire, le développement en série de Taylor correspond à la fonction et l'approximation devient exacte si bien que $\sigma^2 = n\mathrm{Var}[\hat{v}_n] = \mathrm{Var}[g(\mathbf{X})]$. En pratique, σ^2 est estimé en remplaçant $\boldsymbol{\mu}$ par $\hat{\boldsymbol{\mu}}_n$ et $\boldsymbol{\Sigma}_{\mathrm{XX}}$ par la matrice de covariances empiriques notée $\hat{\boldsymbol{\Sigma}}_{\mathrm{XX},n}$.

Souvent, la fonction considérée est un rapport de deux moyennes. Soit dans ce cas

$\mathbf{X} = (X_1, X_2)$, $\boldsymbol{\mu} = (\mu_1, \mu_2)$,

$$\boldsymbol{\Sigma}_{\mathrm{XX}} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix},$$

$\sigma_1^2 = \mathrm{Var}[X_1]$, $\sigma_2^2 = \mathrm{Var}[X_2]$ et $\sigma_{12} = \mathrm{Cov}[X_1, X_2]$. Nous avons $g(\boldsymbol{\mu}) = \mu_1/\mu_2$ et le gradient est $\nabla g(\mu_1, \mu_2) = (1/\mu_2, -\mu_1/\mu_2^2)^{\mathsf{t}}$. La variance, quant à elle, est

$$\sigma^2 \quad = \quad (\sigma_1^2 + \sigma_2^2 \nu^2 - 2\sigma_{12}\nu)/\mu_2^2. \tag{6.2}$$

Pour réduire cette variance, il faut, d'après cette formule, réduire la variance au numérateur et au dénominateur ou augmenter la covariance entre les deux termes.

Nous souhaitons en général augmenter l'*efficacité* de $\hat{\nu}_n$ définie par

$$\mathrm{Eff}[\hat{\nu}_n] = \frac{1}{\mathrm{MSE}[\hat{\nu}_n] C[\hat{\nu}_n]} \tag{6.3}$$

où

$$\mathrm{MSE}[\hat{\nu}_n] = \beta[\hat{\nu}_n] + \mathrm{Var}[\hat{\nu}_n]$$

est l'*erreur quadratique moyenne*,

$$\beta[\hat{\nu}_n] = \mathbb{E}[\hat{\nu}_n] - \nu$$

est son *biais* et $C[\hat{\nu}_n]$ est son coût de calcul moyen par l'ordinateur. Selon cette définition, nous pouvons accroître l'efficacité d'un estimateur en réduisant son biais, sa variance ou son temps de calcul. Dans ce chapitre, nous allons nous concentrer sur la réduction de variance.

6.1 Variables de contrôle

Pour réduire la variance telle que définie précédemment, nous pouvons tenter d'utiliser une ou plusieurs *variables de contrôle* linéaires. Cette technique consiste à ajouter une somme pondérée de variables aléatoires d'espérance nulle à l'estimateur qui nous intéresse. Elle est étudiée pour le cas d'une seule variable aléatoire dans [26, 29] tandis que le cas d'une fonction de plusieurs moyennes, que nous décrivons ici, est présenté dans [25, 27].

Soit $\mathbf{C} \in \mathbb{R}^q$ un vecteur aléatoire dont $\mathbf{c} = \mathbb{E}[\mathbf{C}]$ est connue et calculable efficacement. Soit $\hat{\mathbf{c}}_n$ un estimateur asymptotiquement sans biais de $\mathbf{c}$. L'estimateur avec variables de contrôle, qui remplace simplement $g(\mathbf{X})$ par une autre fonction, est alors

$$\hat{v}_{\mathrm{c},n} = h(\hat{\boldsymbol{\mu}}_n, \hat{\mathbf{c}}_n) = g(\hat{\boldsymbol{\mu}}_n) - \boldsymbol{\beta}^{\mathrm{t}}(\hat{\mathbf{c}}_n - \mathbf{c}) \tag{6.4}$$

où $\boldsymbol{\beta} \in \mathbb{R}^q$ est un vecteur colonne de coefficients à déterminer. Cet estimateur $\hat{v}_{\mathrm{c},n}$, sans biais si $g(\boldsymbol{\mu})$ est linéaire, est une généralisation de l'estimateur plus connu s'appliquant sur un simple scalaire.

Soit

$$\begin{aligned}
\sigma_{\mathrm{c}}^2 \stackrel{\mathrm{def}}{=} \lim_{n\to\infty} n\mathrm{Var}[\hat{v}_{\mathrm{c},n}] &= (\nabla h(\boldsymbol{\mu}, \mathbf{c}))^{\mathrm{t}} \boldsymbol{\Sigma}_{\mathrm{c}} \nabla h(\boldsymbol{\mu}, \mathbf{c}), \\
\text{où } \nabla h(\boldsymbol{\mu}, \mathbf{c}) &= \begin{pmatrix} \nabla g(\boldsymbol{\mu}) \\ -\boldsymbol{\beta} \end{pmatrix} \\
\text{et } \boldsymbol{\Sigma}_{\mathrm{c}} &= \mathrm{Cov}\begin{bmatrix} \mathbf{X} \\ \mathbf{C} \end{bmatrix} = \begin{pmatrix} \boldsymbol{\Sigma}_{\mathrm{XX}} & \boldsymbol{\Sigma}_{\mathrm{XC}} \\ \boldsymbol{\Sigma}_{\mathrm{CX}} & \boldsymbol{\Sigma}_{\mathrm{CC}} \end{pmatrix}.
\end{aligned}$$

Ici, $\boldsymbol{\Sigma}_{\mathrm{XX}} = \mathrm{Cov}[\mathbf{X}]$ est une matrice de $d \times d$, $\boldsymbol{\Sigma}_{\mathrm{CC}} = \mathrm{Cov}[\mathbf{C}]$ est de dimensions $q \times q$ tandis que $\boldsymbol{\Sigma}_{\mathrm{XC}}^{\mathrm{t}} = \boldsymbol{\Sigma}_{\mathrm{CX}} = \mathrm{Cov}[\mathbf{C}, \mathbf{X}]$ est de dimensions $q \times d$. Nous allons considérer que toutes

ces matrices sont définies positives. Alors,

$$\sigma_c^2 \;=\; \sigma^2 - 2\boldsymbol{\beta}^t\boldsymbol{\Sigma}_{CX}\nabla g(\boldsymbol{\mu}) + \boldsymbol{\beta}^t\boldsymbol{\Sigma}_{CC}\boldsymbol{\beta}. \tag{6.5}$$

Selon le choix du vecteur $\boldsymbol{\beta}$, la variance asymptotique peut être réduite ou augmentée. Avec $\boldsymbol{\beta} = \mathbf{0}$, $\sigma_c^2 = \sigma^2$ (aucun changement de la variance). La valeur de $\boldsymbol{\beta}$ minimisant la variance asymptotique est donnée par

$$\boldsymbol{\beta}^* = (\boldsymbol{\Sigma}_{CC})^{-1}\boldsymbol{\Sigma}_{CX}\nabla g(\boldsymbol{\mu}). \tag{6.6}$$

Avec ce vecteur optimal, la variance devient

$$\sigma_c^2 \;=\; (1 - R_{CX}^2)\sigma^2 \tag{6.7}$$

où

$$R_{CX}^2 = \frac{(\nabla g(\boldsymbol{\mu}))^t\boldsymbol{\Sigma}_{CX}^t(\boldsymbol{\Sigma}_{CC})^{-1}\boldsymbol{\Sigma}_{CX}\nabla g(\boldsymbol{\mu})}{\sigma^2}$$

est le *coefficient de détermination* (ou le carré de la corrélation multiple) de $g(\mathbf{X})$ et $\mathbf{C}$. Si $\boldsymbol{\Sigma}_{CC}$ est définie positive, ce que nous avons supposé, $\sigma_c^2 \leq \sigma^2$.

En pratique, $\boldsymbol{\mu}$ et $\boldsymbol{\Sigma}_{CX}$ sont inconnues et remplacées par leurs équivalents empiriques $\hat{\boldsymbol{\mu}}_n$ et $\hat{\boldsymbol{\Sigma}}_{CX,n}$ estimés soit par des expériences pilotes, soit avec les observations employées pour calculer $\hat{v}_{c,n}$. Même si $\boldsymbol{\Sigma}_{CC}$ est parfois connue, l'estimer elle aussi ajoute un contrôle non linéaire à l'estimateur et réduit davantage la variance [34, 45]. Le vecteur de constantes estimé est alors

$$\hat{\boldsymbol{\beta}}_n = (\hat{\boldsymbol{\Sigma}}_{CC,n})^{-1}\hat{\boldsymbol{\Sigma}}_{CX,n}\nabla g(\hat{\boldsymbol{\mu}}_n). \tag{6.8}$$

Si les matrices sont estimées avec les observations utilisées pour estimer v, cela ajoute du biais à l'estimateur $\hat{v}_{c,n}$, mais ce biais est négligeable si n est très grand.

Comme nous l'aborderons à la section 7.1, notre objectif de recherche par rapport à

cette technique consiste à trouver de bonnes variables de contrôle réduisant efficacement la variance.

6.2 Variables aléatoires communes

Nous souhaitons parfois comparer deux ou plusieurs configurations similaires, ne différant par exemple que par un paramètre θ. Soit alors $\mathbf{X}(\theta)$ la valeur de $\mathbf{X}$ conditionnelle à θ, $\boldsymbol{\mu}(\theta) = \mathbb{E}[\mathbf{X}(\theta)]$ le vecteur de mesures de performance conditionnel à θ, et $\nu(\theta) = g(\boldsymbol{\mu}(\theta))$ le résultat de la fonction de plusieurs espérances qui nous intéresse.

Nous souhaitons estimer la différence

$$\Delta = \nu(\theta_2) - \nu(\theta_1)$$

pour $\theta_1 < \theta_2$ et $\delta = \theta_2 - \theta_1$. Soit pour cela $\hat{\nu}_n(\theta, \mathbf{U}) = g(\hat{\boldsymbol{\mu}}_n(\theta, \mathbf{U}))$ un estimateur de $\nu(\theta)$, où $\hat{\boldsymbol{\mu}}_n(\theta, \mathbf{U})$ est un estimateur asymptotiquement sans biais de $\boldsymbol{\mu}(\theta)$ utilisant une séquence de variables aléatoires uniformes $\mathbf{U}$. La différence est estimée par

$$\hat{\Delta}_n = \hat{\nu}_n(\theta_2, \mathbf{U}_2) - \hat{\nu}_n(\theta_1, \mathbf{U}_1)$$

et sa variance est

$$\mathrm{Var}[\hat{\Delta}_n] = \mathrm{Var}[\hat{\nu}_n(\theta_1, \mathbf{U}_1)] + \mathrm{Var}[\hat{\nu}_n(\theta_2, \mathbf{U}_2)] - 2\mathrm{Cov}[\hat{\nu}_n(\theta_1, \mathbf{U}_1), \hat{\nu}_n(\theta_2, \mathbf{U}_2)].$$

Si les deux systèmes sont simulés indépendamment, $\mathbf{U}_1$ et $\mathbf{U}_2$ sont indépendants si bien que la covariance entre les deux estimateurs est nulle et la variance devient

$$\mathrm{Var}[\hat{\Delta}_n] = \mathrm{Var}[\hat{\nu}_n(\theta_1, \mathbf{U}_1)] + \mathrm{Var}[\hat{\nu}_n(\theta_2, \mathbf{U}_2)] \approx 2\mathrm{Var}[\hat{\nu}_n(\theta_1, \mathbf{U}_1)]$$

si θ_1 est assez près de θ_2.

Pour réduire cette variance, nous pouvons utiliser les *variables aléatoires communes*

[10, 41] en fixant $\mathbf{U}_1 = \mathbf{U}_2$. Pour ce faire, les deux systèmes sont simulés avec les mêmes nombres aléatoires, en veillant le plus possible à utiliser les mêmes nombres aux mêmes endroits. Dans le cas de bon nombre de modèles, cela suffit pour induire une corrélation positive entre $\hat{v}_n(\theta_1, \mathbf{U}_1)$ et $\hat{v}_n(\theta_2, \mathbf{U}_2)$ qui réduit $\mathrm{Var}[\hat{\Delta}_n]$.

Il peut être intéressant de combiner cette technique simple avec d'autres techniques telles que les variables de contrôle et la stratification. La *synchronisation*, c'est-à-dire utiliser les nombres aléatoires aux mêmes endroits, n'est également pas toujours facile, car elle peut être effectuée de plusieurs façons dans un modèle complexe.

Nous nous intéressons également à des différences lorsque δ est très petit, par exemple pour estimer une dérivée par la méthode des différences finies

$$\frac{\hat{v}_n(\theta + \delta, \mathbf{U}_2) - \hat{v}_n(\theta, \mathbf{U}_1)}{\delta}.$$

Sans variables aléatoires communes, la variance d'un tel estimateur est approximativement $2\mathrm{Var}[\hat{v}_n(\theta, \mathbf{U})]/\delta^2$ et tend vers l'infini si $\delta \to 0$. Par contre, si nous utilisons les variables aléatoires communes, nous obtenons souvent un estimateur utilisable dont la variance est bornée. Dans certains cas, il est même possible de prouver l'existence d'une telle borne sur la variance en fonction de δ, ce que nous allons tenter de faire dans certains contextes spécifiques (voir section 7.4).

6.3 Variables antithétiques

La méthode des *variables antithétiques*, qui ressemble aux variables aléatoires communes, consiste quant à elle à calculer l'estimateur

$$\hat{v}_{\mathrm{AV},n}(\mathbf{U}) = \frac{\hat{v}_n(\mathbf{U}) + \hat{v}_n(1 - \mathbf{U})}{2}$$

où $\hat{v}_n(\mathbf{U}) = g(\hat{\boldsymbol{\mu}}_n(\mathbf{U}))$ et $\hat{\boldsymbol{\mu}}_n(\mathbf{U})$ est un estimateur de $\boldsymbol{\mu}$ utilisant une séquence $\mathbf{U}$ de variables aléatoires. La variance de cet estimateur est

$$\begin{aligned}
\mathrm{Var}[\hat{v}_{\mathrm{AV},n}(\mathbf{U})] &= \frac{1}{4}(\mathrm{Var}[\hat{v}_n(U)] + \mathrm{Var}[\hat{v}_n(1-U)] + 2\mathrm{Cov}[\hat{v}_n(\mathbf{U}), \hat{v}_n(1-\mathbf{U})]) \\
&= \frac{\mathrm{Var}[\hat{v}_n] + \mathrm{Cov}[\hat{v}_n(\mathbf{U}), \hat{v}_n(1-\mathbf{U})]}{2} \\
&< \mathrm{Var}[\hat{v}_n]
\end{aligned}$$

si

$$\mathrm{Cov}[\hat{v}_n(\mathbf{U}), \hat{v}_n(1-\mathbf{U})] < 0.$$

Pour que la technique fonctionne, il doit ainsi exister une corrélation négative entre $\hat{v}_n(\mathbf{U})$ et $\hat{v}_n(1-\mathbf{U})$. Intuitivement, les événements désastreux pendant la simulation calculant $\hat{v}_n(\mathbf{U})$ doivent être compensés par des événements heureux dans la simulation pour $\hat{v}_n(1-\mathbf{U})$.

L'application de cette technique est très simple, mais il peut être intéressant de tenter de la combiner avec d'autres techniques telles que les variables de contrôle ou la stratification. Nous devons également faire face aux mêmes problèmes de synchronisation qu'avec les variables aléatoires communes.

6.4 Monte Carlo conditionnel

La méthode *Monte Carlo conditionnel* remplace un estimateur X par une espérance conditionnelle

$$X_{\mathrm{CMC}} = \mathbb{E}[X \mid Y].$$

La variance de cet estimateur est

$$\mathrm{Var}[X_{\mathrm{CMC}}] = \mathrm{Var}[X] - \mathbb{E}[\mathrm{Var}[X \mid Y]] \leq \mathrm{Var}[X].$$

Cela signifie que plus la variance conditionnelle de X étant donné Y est élevée, plus la technique réduit la variance. En d'autres mots, moins Y contient d'informations pour calculer X, plus la variance est réduite. La variable aléatoire Y pourrait être remplacée par un vecteur sans changer l'expression de $\mathrm{Var}[X_{\mathrm{CMC}}]$. Nous pouvons aussi appliquer cette technique pour remplacer un vecteur $\mathbf{X}$ par une espérance conditionnelle $\mathbb{E}[\mathbf{X} \mid Y]$, mais aucune expression n'est disponible pour la matrice de covariances résultante.

Comme nous le verrons à la section 7.5, nous allons tenter de combiner cette technique aux variables aléatoires communes pour estimer des dérivées.

6.5 Échantillonnage stratégique

L'*échantillonnage stratégique* consiste à remplacer la fonction de densité de l'estimateur afin de nous concentrer sur les endroits où la variance est la plus élevée. Supposons que le vecteur $\mathbf{X}$ a une densité conjointe donnée par $\pi(\mathbf{X})$ et que nous voulons calculer $g(\mathbf{X})$. Pour appliquer la méthode, nous générons $\mathbf{X}$ par rapport à la densité $f(\mathbf{X})$ au lieu de la densité $\pi(\mathbf{X})$ et calculons l'estimateur

$$X_{\mathrm{IS}} = g(\mathbf{X})\frac{\pi(\mathbf{X})}{f(\mathbf{X})} = g(\mathbf{X})L(\mathbf{X}),$$

où

$$L(\mathbf{X}) \stackrel{\text{def}}{=} \frac{\pi(\mathbf{X})}{f(\mathbf{X})}$$

est appelé le *rapport de vraisemblance*. Malheureusement, la densité de $\mathbf{X}$ est souvent trop complexe pour être calculée directement. Dans certains cas, nous pouvons récrire $g(\mathbf{X})$ comme $h(\mathbf{Y})$ où $\mathbf{Y} = (Y_1, \ldots, Y_d)$ est un vecteur de variables aléatoires indépendantes entre elles. La densité $\pi_i(Y_i)$ est alors remplacée par $f_i(Y_i)$ et le rapport de vraisemblance devient

$$L(\mathbf{Y}) = \frac{\pi_1(Y_1)\cdots\pi_d(Y_d)}{f_1(Y_1)\cdots f_d(Y_d)}.$$

La difficulté de cette technique consiste à trouver un bon changement de densité qui réduit effectivement la variance bornée par

$$\mathrm{Var}[X_{\mathrm{IS}}] \leq \rho \mathrm{Var}[X] \qquad \text{si} \qquad L(\mathbf{Y}) \leq \rho .$$

Une façon de démontrer que la variance est réduite consiste alors à prouver que le rapport de vraisemblance ne peut pas excéder 1 si $X_{\mathrm{IS}} \neq 0$.

Nous pourrions tenter d'appliquer cette technique pour estimer la probabilité que le niveau de service sur une longue période soit inférieur à un seuil (voir section 7.6).

6.6 Méthodes quasi-Monte Carlo randomisées

Les *méthodes quasi-Monte Carlo* consistent à remplacer les variables aléatoires suivant la loi uniforme par des variables déterministes distribuées de façon plus uniforme que des variables aléatoires. Le résultat de la simulation peut être considéré comme une fonction $\hat{\boldsymbol{\mu}}(\mathbf{U})$ d'un point $\mathbf{U}$ sur l'hypercube $[0,1)^t$, avec $\boldsymbol{\mu} = \mathbb{E}[\hat{\boldsymbol{\mu}}(\mathbf{U})]$. La méthode Monte Carlo génère n points sur l'hypercube de façon aléatoire et uniforme pour ensuite calculer n copies de $\hat{\boldsymbol{\mu}}(\mathbf{U}_i)$. Dans les méthodes quasi-Monte Carlo [31, 39], les n valeurs aléatoires sont remplacées par un ensemble de points déterministe construit de façon à être très uniformément distribué sur l'hypercube.

Mais la variance ainsi obtenue est nulle si bien qu'il devient difficile d'estimer l'erreur d'estimation. Pour résoudre ce problème, il suffit de randomiser l'ensemble de points à l'aide d'une méthode qui conserve la grande uniformité des points et qui fait en sorte que chaque point est distribué sur l'hypercube $[0,1)^t$ selon la loi uniforme. Si nous générons m randomisations d'un même ensemble de n points, nous obtenons m copies indépendantes de l'estimateur

$$\hat{\boldsymbol{\mu}}_{\mathrm{RQMC},i} = \frac{1}{n} \sum_{j=1}^{n} \hat{\boldsymbol{\mu}}(\mathbf{U}_{i,j})$$

où $\mathbf{U}_{i,j}$ est le point j de la randomisation i. Avec ces randomisations, nous pouvons calculer une moyenne globale

$$\hat{\boldsymbol{\mu}}_{\text{RQMC}} = \frac{1}{m}\sum_{i=1}^{m}\hat{\boldsymbol{\mu}}_{\text{RQMC},i} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\hat{\boldsymbol{\mu}}(\mathbf{U}_{i,j})$$

et la matrice de covariances empiriques estimant $\text{Cov}[\hat{\boldsymbol{\mu}}_{\text{RQMC},i}]$. Il existe de nombreuses façons de construire un ensemble de points, notamment les règles de réseaux, les réseaux digitaux, etc., ainsi que plusieurs techniques de randomisation.

Cette méthode fonctionne bien seulement lorsque la dimension du problème est petite. Pour appliquer cette méthode à un problème complexe comme la simulation d'un centre de contacts, il nous faut par conséquent réduire la dimension effective du problème (voir section 7.7).

6.7 Stratification

Cette technique, qui existe déjà depuis quelques années [17], consiste à subdiviser l'échantillon servant à estimer une moyenne en plusieurs strates dans lesquelles la variance est réduite. Nous la généralisons ici pour le cas d'une fonction de plusieurs moyennes, ce qui est innovateur. Nous allons aussi traiter le cas où nous stratifions par rapport à une variable aléatoire continue.

Soit S une variable aléatoire ayant une grande corrélation avec $\mathbf{X}$ ou, mieux encore, avec $g(\mathbf{X})$. En supposant que S est discrète sur un support fini $\{1,\dots,m\}$ et a une fonction de masse $p_s = \mathbb{P}[S=s]$, nous pouvons utiliser la probabilité totale pour récrire $\boldsymbol{\mu}$ comme

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}] = \sum_{s=1}^{m} p_s \boldsymbol{\mu}_s \qquad \text{avec} \qquad \boldsymbol{\mu}_s = \mathbb{E}[\mathbf{X} \mid S=s]$$

l'espérance de $\mathbf{X}$ étant donné que $S=s$. La *stratification* consiste à estimer $\boldsymbol{\mu}_s$ de façon indépendante pour chaque valeur de $S=s$, appelée *strate*, pour ensuite obtenir une estimation de $\boldsymbol{\mu}$.

Supposons maintenant que nous stratifions par rapport à un vecteur aléatoire $\mathbf{U} \in [0,1)^t$ dont les éléments peuvent être utilisés pour générer des variables aléatoires de la simulation. Nous pouvons dans ce cas diviser l'hypercube en m sous-cubes, avec p_s le volume du sous-cube s. Pour échantillonner dans la strate s, il suffit alors de générer $\mathbf{U}$ uniformément dans le sous-cube s, de fixer $S = s$ et d'utiliser $\mathbf{U}$ pour la simulation.

Soit n_s le nombre d'observations disponibles pour $S = s$ et

$$n = \sum_{s=1}^{m} n_s$$

le nombre total d'observations. Soit $\mathbf{X}_{s,j} \in \mathbb{R}^d$ l'observation j dans la strate s, pour $j = 1, \ldots, n_s$ et $s = 1, \ldots, m$. L'estimateur stratifié de $\boldsymbol{\mu}$ est alors

$$\bar{\mathbf{X}}_{\text{strat},n,m} = \sum_{s=1}^{m} p_s \hat{\boldsymbol{\mu}}_s \qquad \text{où} \qquad \hat{\boldsymbol{\mu}}_s = \frac{1}{n_s} \sum_{j=1}^{n_s} \mathbf{X}_{s,j} \tag{6.9}$$

est un estimateur de $\boldsymbol{\mu}_s$. Soit $\boldsymbol{\Sigma}_{\text{strat}} \stackrel{\text{def}}{=} n\text{Cov}[\bar{\mathbf{X}}_{\text{strat},n,m}]$ la matrice de covariances de l'estimateur stratifié et $\boldsymbol{\Sigma}_s = \text{Cov}[\mathbf{X} \mid S = s]$ la matrice des covariances conditionnelles à la strate s. Alors,

$$\boldsymbol{\Sigma}_{\text{strat}} = n \sum_{s=1}^{m} \frac{p_s^2 \boldsymbol{\Sigma}_s}{n_s}. \tag{6.10}$$

Nous avons défini $\boldsymbol{\Sigma}_{\text{strat}}$ de cette façon afin qu'elle soit comparable avec $\boldsymbol{\Sigma}_{\text{XX}}$.

Pour estimer $g(\boldsymbol{\mu})$ avec la stratification, nous calculons $\hat{v}_{\text{strat},n,m} = g(\bar{\mathbf{X}}_{\text{strat},n,m})$, c'est-à-dire la fonction avec l'estimateur stratifié (6.9). La variance asymptotique de cet estimateur est

$$\begin{aligned}
\sigma^2_{\text{strat}} &\stackrel{\text{def}}{=} \lim_{n\to\infty} n\text{Var}[g(\bar{\mathbf{X}}_{\text{strat},n,m})] && (6.11)\\
&= (\nabla g(\boldsymbol{\mu}))^{\text{t}} \boldsymbol{\Sigma}_{\text{strat}} \nabla g(\boldsymbol{\mu}) && (6.12)\\
&= n \sum_{s=1}^{m} \frac{p_s^2 (\nabla g(\boldsymbol{\mu}))^{\text{t}} \boldsymbol{\Sigma}_s \nabla g(\boldsymbol{\mu})}{n_s}
\end{aligned}$$

$$= \quad n \sum_{s=1}^{m} \frac{p_s^2 \sigma_s^2}{n_s}$$

où

$$\sigma_s^2 \stackrel{\text{def}}{=} (\nabla g(\boldsymbol{\mu}))^{\mathrm{t}} \boldsymbol{\Sigma}_s \nabla g(\boldsymbol{\mu})$$

est la contribution de la strate s à la variance globale. Avec cette définition, si $g(\boldsymbol{\mu}) = \mu_i$, $\sigma^2_{\text{strat}} = \text{Var}[X_i]$, $\sigma_s^2 = \text{Var}[X_i \mid S = s] = \text{Var}[X_{s,j}]$ et nous retrouvons la formule habituelle pour la variance d'un estimateur stratifié [17].

En supposant qu'il est possible de générer $\mathbf{X}$ conditionnellement à $S = s$ pour un s choisi, nous pouvons ajuster les valeurs de n_s avant la simulation. Il existe pour cela plusieurs mécanismes d'allocation. En particulier, l'*allocation proportionnelle* fixe $n_s = np_s$. La variance devient

$$\boldsymbol{\Sigma}_{\text{stratp}} \stackrel{\text{def}}{=} n \sum_{s=1}^{m} \frac{p_s^2 \boldsymbol{\Sigma}_s}{np_s} = \sum_{s=1}^{m} p_s \boldsymbol{\Sigma}_s, \qquad \sigma^2_{\text{stratp}} = n \sum_{s=1}^{m} \frac{p_s^2 \sigma_s^2}{np_s} = \sum_{s=1}^{m} p_s \sigma_s^2. \tag{6.13}$$

Nous pouvons également rechercher l'*allocation optimale* visant à minimiser σ^2_{strat}. En utilisant un Lagrangien pour traiter la contrainte $n = \sum_{s=1}^{m} n_s$, nous obtenons

$$n_s = \frac{np_s\sigma_s}{\bar{\sigma}} \qquad \text{avec} \qquad \bar{\sigma} = \sum_{s=1}^{m} p_s \sigma_s. \tag{6.14}$$

La variance avec cette allocation est

$$\sigma^2_{\text{strato}} = n\bar{\sigma} \sum_{s=1}^{m} \frac{p_s^2 \sigma_s^2}{np_s\sigma_s} = \bar{\sigma} \sum_{s=1}^{m} p_s \sigma_s = \bar{\sigma}^2. \tag{6.15}$$

Les valeurs de σ_s^2, inconnues au départ, doivent être estimées par des expériences pilotes avant de pouvoir estimer les valeurs de n_s optimales. Chaque valeur de n_s obtenue est ensuite arrondie à l'entier le plus près et fixée à 2 si elle est inférieure à ce nombre.

La matrice de covariances peut être décomposée comme suit.

$$\begin{aligned} & \mathrm{Cov}[\mathbf{X}] \\ = \;& \mathbb{E}[\mathrm{Cov}[\mathbf{X} \mid S]] + \mathrm{Cov}[\mathbb{E}[\mathbf{X} \mid S]] \\ = \;& \sum_{s=1}^{m} p_s \boldsymbol{\Sigma}_s + \sum_{s=1}^{m} p_s (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}} \\ = \;& \boldsymbol{\Sigma}_{\text{stratp}} + \sum_{s=1}^{m} p_s (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}}. \end{aligned} \tag{6.16}$$

En remplaçant $\boldsymbol{\Sigma}_{\mathrm{XX}}$ par (6.16) dans l'expression de σ^2, nous pouvons décomposer la variance asymptotique de la façon suivante.

$$\begin{aligned} \sigma^2 &= (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} \boldsymbol{\Sigma}_{\mathrm{XX}} \nabla g(\boldsymbol{\mu}) \\ &= \sum_{s=1}^{m} p_s (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} \boldsymbol{\Sigma}_s \nabla g(\boldsymbol{\mu}) + \sum_{s=1}^{m} p_s (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}} \nabla g(\boldsymbol{\mu}) \\ &= \sum_{s=1}^{m} p_s \sigma_s^2 + \sum_{s=1}^{m} p_s (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}} \nabla g(\boldsymbol{\mu}) \\ &= \sigma^2_{\text{stratp}} + \sum_{s=1}^{m} p_s (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}} \nabla g(\boldsymbol{\mu}) \qquad (6.17) \\ &= \sigma^2_{\text{strato}} + \sum_{s=1}^{m} p_s (\sigma_s - \bar{\sigma})^2 \\ &\quad + \sum_{s=1}^{m} p_s (\nabla g(\boldsymbol{\mu}))^{\mathsf{t}} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^{\mathsf{t}} \nabla g(\boldsymbol{\mu}). \qquad (6.18) \end{aligned}$$

Cette décomposition montre que l'allocation proportionnelle élimine la variance causée par la différence entre les moyennes dans les strates. L'allocation optimale, quant à elle, élimine aussi la différence de variances entre les strates. De plus, si $g(\boldsymbol{\mu}) = \mu_i$, nous retrouvons la décomposition habituelle présentée dans [17].

Nous allons discuter de l'application de cette technique à la section 7.2 et de sa combinaison avec les variables de contrôle à la section 7.3.

6.8 Scission

Il arrive que le nombre nécessaire de réplications soit élevé en raison d'une période de l'horizon dans laquelle la variance est très élevée. Simuler plusieurs réplications est alors coûteux inutilement pour les périodes dans lesquelles la variance est faible. Si nous comparons plusieurs configurations, il se peut aussi que le comportement du système simulé ne diffère pas beaucoup d'une valeur à l'autre des paramètres testés. Dans ce cas également, du travail de simulation est fait inutilement.

Pour réduire le travail de simulation dans ces deux situations, nous pouvons appliquer la scission (*splitting* en Anglais) de la façon suivante : à un certain point pendant la simulation, une réplication est dupliquée et chaque copie est simulée en parallèle, indépendamment. Chacune de ces copies peut être scindée à son tour, ce qui produit un arbre de simulation. Inversement, si à un moment donné, plusieurs réplications partagent un état identique, les copies peuvent être tuées afin d'élaguer l'arbre et d'éviter du travail inutile. Cette technique a par exemple été appliquée à des chaînes de Markov pour estimer la probabilité d'événements rares [20, 38] ainsi qu'à des cas où des décisions binaires pouvaient être prises à des moments aléatoires pendant la simulation [42]. Nous allons l'appliquer ici à un système dont l'état est beaucoup trop complexe pour être analysé comme une chaîne de Markov et où les décisions conditionnant la scission ne sont pas binaires. Ces deux applications sont nouvelles et prometteuses. Nous traiterons de nos objectifs de recherche par rapport à ces méthodes dans la section 7.8 du chapitre suivant.

6.8.1 Périodes avec variance élevée

Nous allons supposer que nous simulons le modèle sur un horizon fini de durée T et divisé en P périodes principales. La valeur de T et la durée des périodes peuvent être aléatoires, mais P est fixe. Soit alors $\mathbf{Y} = (Y_1, \ldots, Y_p)$ un vecteur aléatoire avec une composante par période. Soit $\mathbf{\Sigma}_{\mathrm{YY}} = \mathrm{Cov}[\mathbf{Y}]$ la matrice de covariances de $\mathbf{Y}$ et soit $\sigma_p^2 = \mathrm{Var}[Y_p]$ la variance pour la période p. Nous allons supposer que la variance dif-

fère fortement entre les périodes et que les covariances sont petites. La valeur globale couvrant tout l'horizon est donnée par

$$Y = \sum_{p=1}^{P} w_p Y_p$$

où w_p est la pondération associée à la période p ; souvent $w_p = 1$.

Nous voulons effectuer davantage de réplications dans les périodes p pour lesquelles σ_p^2 est élevée. Pour cela, nous allons scinder et fusionner uniquement en début de périodes.

La section suivante propose une méthode pour effectuer la simulation avec scission. Nous étudions ensuite le comportement de la variance lorsque le nombre de réplications peut différer d'une période à l'autre. Nous proposons finalement une technique pour fixer le nombre de réplications parallèles par période.

6.8.1.1 Simulation d'une réplication avec scission

Soit $\mathbf{n} = (n_1, \ldots, n_P)$ le nombre de réplications parallèles pour chacune des P périodes, que nous allons considérer fixe pour le moment. Sans scission, nous aurions $n_1 = \cdots = n_P$. Soit maintenant G_p la loi de probabilité caractérisant l'état du système au début de la période p. Pour chaque période $p = 1, \ldots, P$, nous voudrions idéalement générer un état depuis cette distribution G_p, simuler la période en partant de cet état et répéter ce processus n_p fois. Malheureusement, nous ne connaissons pas la distribution G_p si bien que nous devons l'approximer de la façon suivante.

Supposons que l'état du système au début de la période 1 est déterministe ou que nous connaissons G_1. Ceci est une hypothèse raisonnable puisque le système est souvent vide au début de cette période. Nous pouvons alors démarrer n_1 réplications parallèles et indépendantes à partir du début de la période 1 et les terminer au début de la période 2. Cela nous fournit n_1 états finaux qui sont utilisés pour approximer G_2.

De façon générale, à la fin de la période $p-1$, pour $p > 1$, nous disposons d'un

échantillon de n_{p-1} états finaux permettant d'estimer G_p. Pour générer un état de début pour la période p, nous choisissons simplement un état de fin de la période $p-1$ et le clonons ; ce processus est répété n_p fois pour obtenir l'état initial de toutes les réplications parallèles nécessaires. Chaque réplication peut alors être simulée indépendamment. La même technique est utilisée pour chacune des périodes à simuler.

Il existe plusieurs algorithmes pour effectuer le choix des états initiaux de la période $p > 1$ à partir des états finaux de la période $p-1$. Dans la *scission fixe*, chacune des n_{p-1} réplications, pour $p > 1$, est scindée c_p fois, d'où $n_p = c_p n_{p-1}$ si c_p est un entier. Si c_p n'est pas entier, nous pouvons, pour chaque réplication parallèle, faire $\lfloor c_p \rfloor$ copies ainsi qu'une copie additionnelle avec probabilité $c_p - \lfloor c_p \rfloor$. Si $0 < c_p < 1$, chaque réplication parallèle est tuée avec probabilité c_p.

Puisque nous avons fixé n_p, nous allons plutôt utiliser la stratégie *effort fixe* qui consiste à choisir n_p états initiaux pour la période p parmi l'ensemble des n_{p-1} états finaux pour la période $p-1$. Pour ce faire, nous pouvons utiliser les stratégies d'affectation aléatoire ou fixe [20]. L'*affectation aléatoire* choisit n_p états initiaux avec remplacement parmi les n_{p-1} états finaux. L'*affectation fixe* choisit quant à elle $d_p = n_p \bmod n_{p-1}$ états finaux parmi les n_{p-1} états disponibles, cette fois-ci sans remplacement, et les clone $c_p + 1$ fois, où $c_p = \lfloor n_p / n_{p-1} \rfloor$. Les autres états sont clonés c_p fois. L'affectation fixe équivaut à un échantillonnage stratifié des états finaux.

6.8.1.2 Variance obtenue

À la fin du processus de simulation, nous disposons, pour $p = 1, \ldots, P$, de n_p observations $Y_{1,p}, \ldots, Y_{n_p,p}$. Si l'état au début de la période 1 est déterministe, les observations $Y_{j,1}$ sont indépendantes entre elles puisqu'elles partagent ce même état. Les observations $Y_{j,p}$ provenant de la scission de la réplication parallèle j' partagent également un même état initial et sont donc indépendantes conditionnellement à cet état du début de la période p, mais sans ce conditionnement, deux observations quelconques sont généralement dépendantes.

Malgré la dépendance des observations, nous pouvons calculer une moyenne

$$\hat{Y}_p = \frac{1}{n_p} \sum_{j=1}^{n_p} Y_{j,p}$$

pour la période p. Nous pouvons alors obtenir une moyenne globale

$$\hat{Y} = \sum_{p=1}^{P} w_p \hat{Y}_p.$$

La variance de cette moyenne est difficile à estimer, car

$$\mathrm{Var}[\hat{Y}] = \mathrm{Var}\left[\sum_{p=1}^{P} w_p \hat{Y}_p\right] = \sum_{p_1=1}^{P} \sum_{p_2=1}^{P} w_{p_1} w_{p_2} \mathrm{Cov}[\hat{Y}_{p_1}, \hat{Y}_{p_2}]$$

où

$$\begin{aligned} \mathrm{Cov}[\hat{Y}_{p_1}, \hat{Y}_{p_2}] &= \mathrm{Cov}\left[\frac{1}{n_{p_1}} \sum_{j=1}^{n_{p_1}} Y_{j,p_1}, \frac{1}{n_{p_2}} \sum_{j=1}^{n_{p_2}} Y_{j,p_2}\right] \\ &= \frac{1}{n_{p_1} n_{p_2}} \sum_{j_1=1}^{n_{p_1}} \sum_{j_2=1}^{n_{p_2}} \mathrm{Cov}[Y_{j_1,p_1}, Y_{j_2,p_2}] \end{aligned}$$

où $\mathrm{Cov}[Y_{j_1,p_1}, Y_{j_2,p_2}]$ est la covariance entre l'observation j_1 recueillie durant la période p_1 et l'observation j_2 obtenue durant la période p_2. Cette dernière somme comporte beaucoup trop de termes, surtout si n_p est grand. Pour estimer $\mathrm{Var}[\hat{Y}]$, nous pouvons tenter de négliger les dépendances entre réplications parallèles, c'est-à-dire considérer que $\mathrm{Cov}[Y_{j_1,p_1}, Y_{j_2,p_2}] = 0$ si $j_1 \neq j_2$. Ceci est raisonnable seulement si les périodes sont longues. Supposons d'abord que $n_{p_1} = n_{p_2} = n$. Si les observations $Y_{1,p_1}, \ldots, Y_{n,p_1}$ sont indépendantes et identiquement distribuées (i.i.d.) et les observations $Y_{1,p_2}, \ldots, Y_{n,p_2}$ sont aussi i.i.d.,

$$\mathrm{Cov}[\hat{Y}_{p_1}, \hat{Y}_{p_2}] \approx \frac{1}{n^2} \sum_{j=1}^{n} \mathrm{Cov}[Y_{j,p_1}, Y_{j,p_2}] = \frac{\sigma_{p_1,p_2}}{n}.$$

Si $n_{p_1} \neq n_{p_2}$, nous prenons de façon heuristique les $n = \min\{n_{p_1}, n_{p_2}\}$ premières ob-

servations pour p_1 et p_2, ce qui semble raisonnable si les observations sont i.i.d. Nous obtenons ainsi l'approximation

$$\mathrm{Cov}[\hat{Y}_{p_1},\hat{Y}_{p_2}] \approx \frac{1}{n_{p_1}n_{p_2}} \sum_{j=1}^{\min\{n_{p_1},n_{p_2}\}} \mathrm{Cov}[Y_{j,p_1},Y_{j,p_2}] = \frac{\sigma_{p_1,p_2}}{\max\{n_{p_1},n_{p_2}\}}.$$

Selon cette dernière équation, la méthode réduit la covariance entre deux périodes quelconque par un facteur dépendant du nombre de réplications parallèles. Si $p = p_1 = p_2$, nous obtenons bel et bien

$$\mathrm{Var}[\hat{Y}_p] = \sigma_p^2/n_p.$$

La variance globale, avec cette approximation, est

$$\mathrm{Var}[\hat{Y}] \approx \sum_{p_1=1}^{P}\sum_{p_2=1}^{P} \frac{w_{p_1}w_{p_2}\sigma_{p_1,p_2}}{\max\{n_{p_1},n_{p_2}\}}. \tag{6.19}$$

Une autre façon d'estimer la variance consiste à répéter l'expérience complète plusieurs fois. À chaque macro-réplication, nous obtenons alors une moyenne $\hat{Y}_p$ locale, une moyenne $\hat{Y}$ globale et les moyennes sont indépendantes entre les macro-réplications. Nous pouvons alors estimer les variances et les covariances de façon habituelle. Cette méthode permettra d'évaluer si notre approximation précédente de la variance donne de bons résultats. Par contre, pour un même budget de simulation, cette méthode nous oblige à réduire les valeurs de n_p, ce qui réduit la précision de l'estimation des lois G_p.

6.8.1.3 Ajustement optimal des valeurs de n_p

Nous souhaitons minimiser la variance donnée par (6.19) sous la contrainte que $\sum_{p=1}^{P} n_p = n$. Pour ce faire, nous considérons d'abord que $n_p \in \mathbb{R}$ et nous convertissons l'équation en incorporant la contrainte sous la forme d'un Lagrangien

$$\sum_{p=1}^{P}\frac{w_p^2\sigma_p^2}{n_p} + 2\sum_{p_1=1}^{P-1}\sum_{p_2=p_1+1}^{P}\frac{w_{p_1}w_{p_2}\sigma_{p_1,p_2}}{\max\{n_{p_1},n_{p_2}\}} - \lambda\left(n - \sum_{p=1}^{P} n_p\right). \tag{6.20}$$

Cette fonction est discontinue et non dérivable partout où $n_{p_1} \neq n_{p_2}$ pour $p_1 \neq p_2$. Même sans ces discontinuités, cette dérivée, qui doit tenir compte d'un maximum, est trop complexe pour qu'il soit possible de trouver une expression analytique pour n_p.

De façon très heuristique, nous pouvons par contre négliger les covariances entre les périodes pour obtenir une expression pour les n_p. La fonction à minimiser, avec le Lagrangien, devient

$$\sum_{p=1}^{P} \frac{w_p^2 \sigma_p^2}{n_p} - \lambda \left(n - \sum_{p=1}^{P} n_p \right). \tag{6.21}$$

La dérivée partielle de cette expression par rapport à n_p est

$$-\frac{w_p^2 \sigma_p^2}{n_p^2} + \lambda.$$

En mettant ces dérivées partielles à 0, nous obtenons

$$\lambda = \frac{w_p^2 \sigma_p^2}{n_p^2} \; \forall \; p = 1, \ldots, P,$$

d'où

$$n_p = \frac{w_p \sigma_p}{\sqrt{\lambda}}$$

si bien que $n_p \propto w_p \sigma_p$. En dérivant (6.21) par rapport à λ et en mettant cette dérivée à 0, nous obtenons

$$n = \sum_{p=1}^{P} n_p.$$

Alors,

$$n = \sum_{p=1}^{P} \frac{w_p \sigma_p}{\sqrt{\lambda}}$$

si bien que

$$\bar{\sigma} = \sqrt{\lambda} = \frac{1}{n} \sum_{p=1}^{P} w_p \sigma_p.$$

Ainsi, $n_p = w_p \sigma_p / \bar{\sigma}$, où $\bar{\sigma}$ est une constante de normalisation faisant en sorte que la

somme des n_p donne n. Nous constatons alors que trouver les valeurs optimales de n_p est très semblable à l'allocation optimale dans le cas de la stratification, que nous avons décrite à la section 6.7.

Puisque nous ne connaissons pas les variances σ_p^2 en pratique, il nous faut les estimer par un certain nombre d'expériences pilotes.

6.8.2 Calcul de sous-gradients

Les sous-gradients peuvent être utilisés en optimisation pour générer des coupes dans le but d'approximer les contraintes de niveaux de service, qui sont non linéaires et stochastiques, avec des contraintes linéaires [3, 6, 14]. Un sous-gradient peut aussi servir à guider une recherche par voisinage vers un optimum local. Soit $\boldsymbol{\mu}(\mathbf{N})$ (abus de notation) où $\mathbf{N}$ est une matrice de $I \times P$ avec I correspondant au nombre de paramètres par période et P, au nombre de périodes principales formant l'horizon de simulation. Par exemple, I pourrait correspondre au nombre de groupes d'agents dans un centre de contacts.

Le sous-gradient que nous allons considérer se définit comme une matrice de $I \times P$ dont l'élément $S_{i,p}$ est donné par

$$S_{i,p} = \frac{g(\boldsymbol{\mu}(\mathbf{N} + \delta_{i,p}\mathbf{e}_{i,p})) - g(\boldsymbol{\mu}(\mathbf{N}))}{\delta_{i,p}}.$$

Ici, $\mathbf{e}_{i,p}$ est une matrice de $I \times P$ dont l'élément (i,p) est 1 et tous les autres éléments sont 0. Dans le cas d'un centre de contacts, cela correspond à ajouter un agent dans le groupe i durant la période p. Dans le cas où l'élément (i,p) de $\mathbf{N}$ est une valeur entière, $\delta_{i,p}$ doit être entier lui aussi. Sinon, $\delta_{i,p}$ peut être près de 0 afin d'approximer la dérivée partielle par la méthode des différences finies. Souvent, $\delta_{i,p} = \delta$ pour un sous-gradient.

La façon classique de calculer un tel sous-gradient est de simuler n réplications pour $\boldsymbol{\mu}(\mathbf{N})$ et n autres réplications pour chaque valeur de $\boldsymbol{\mu}(\mathbf{N} + \delta_{i,p}\mathbf{e}_{i,p})$, ce qui nécessite nIP réplications au total. Utiliser les variables aléatoires communes permet de réduire la variance sur les composantes du sous-gradient, mais le temps de calcul n'en demeure

pas moins très élevé. Nous pouvons alors penser à éviter de refaire des calculs coûteux en stockant les variables aléatoires générées dans un cache. Bien entendu, un tel cache peut augmenter la vitesse mais au détriment de l'utilisation de la mémoire.

Idéalement, nous aimerions utiliser la scission pour obtenir un estimateur fonctionnel $\boldsymbol{\mu}(\mathbf{N})$ pour toute valeur de $\mathbf{N}$, mais le domaine de $\mathbf{N}$ est habituellement infini. Même si chaque élément de $\mathbf{N}$ est borné par U, cela nous laisse $I * P * U$ valeurs possibles de $\mathbf{N}$. Heureusement, il n'est habituellement pas nécessaire de disposer de toutes ces valeurs de $\boldsymbol{\mu}(\mathbf{N})$; certaines valeurs particulières comme celles pour les sous-gradients suffisent.

Nous allons donc développer une méthode de scission pour estimer un nombre N relativement petit de valeurs $\boldsymbol{\mu}(\mathbf{N}')$ simultanément, par exemple toutes les IP valeurs de $\boldsymbol{\mu}(\mathbf{N} + \delta_{i,p}\mathbf{e}_{i,p})$ pour $i = 1,\ldots,I$ et $p = 1,\ldots,P$. Notre méthode simule toujours une réplication dite *mère* avec la matrice de paramètres de base fixée à $\mathbf{N}$. À certains moments pendant la simulation, le comportement du système dépend de la valeur de $\mathbf{N}$. Un point de scission est un moment à partir duquel le comportement diffère si $\mathbf{N}$ est remplacé par $\mathbf{N}'$. Par exemple, si $\mathbf{N}$ correspond au nombre d'agents dans un centre de contacts, les décisions prises par le routeur sont affectées par $\mathbf{N}$. Lorsqu'un point de scission est rencontré, modifier le nombre d'agents dans un groupe pour une période donnée va affecter comment le prochain contact sera traité par le routeur.

Lorsqu'un tel point de scission est rencontré, il se crée au plus N réplications filles. Par contre, une réplication fille, avec paramètres $\mathbf{N}'$, est produite seulement si une réplication avec paramètres $\mathbf{N}'$ n'est pas déjà en cours de simulation.

À partir du moment où une première réplication fille est créée, le système s'arrête à intervalles réguliers pour comparer l'état des réplications filles avec la réplication mère. Si deux états sont identiques, la réplication fille est tuée et la simulation se poursuit. Par contre, des compteurs statistiques doivent être maintenus pour chaque réplication fille, y compris celles qui ont été tuées, car en général, la valeur de ces compteurs dépend de la trajectoire suivie par toute la réplication. Tout ce processus de scission et de fusion peut être répété plusieurs fois, indépendamment, afin d'accroître la précision d'estimation.

Dans le cas d'un sous-gradient pour un centre de contacts, une décision peut engendrer au plus I réplications filles, c'est-à-dire une par groupe d'agents. Il est également possible d'omettre les comparaisons d'états en considérant que la transformation de $\mathbf{N}$ à $\mathbf{N} + \delta_{i,p}\mathbf{e}_{i,p}$ est locale à la période p. En effet, aucune scission n'a lieu avant cette période tandis que la comparaison de l'état peut ne débuter qu'à la période $p+1$. À la limite, les réplications filles peuvent être tuées arbitrairement à la fin de la période $p+q$, avec $q > 1$, là où le changement de $\mathbf{N}$ n'a qu'un effet minime.

CHAPITRE 7

OBJECTIFS DE RECHERCHE POUR AMÉLIORER L'EFFICACITÉ

7.1 Recherche de bonnes variables de contrôle

Bien que les variables de contrôle, que nous avons décrites de façon générale à la section 6.1, soient très connues en simulation, il n'est pas toujours simple de les appliquer pour un modèle général estimant plusieurs mesures de performance. Dans ce contexte, deux stratégies sont possibles : appliquer plusieurs variables de contrôle à tous les estimateurs calculés ou appliquer des variables de contrôle différentes à chaque estimateur. La première stratégie favorise la simplicité d'implantation au détriment du coût de calcul tandis que la seconde, pour laquelle nous avons opté, maximise la réduction de la variance.

Dans le cas de l'estimation du niveau de service, la première variable de contrôle que nous avons testée dans [36] est le nombre d'arrivées. Pour un estimateur concernant les contacts de type k arrivés pendant la période p, nous utilisons le nombre $A_{k,p}$ d'arrivées de type k pendant la période p comme variable de contrôle afin de maximiser la corrélation entre la variable de contrôle et la mesure testée.

Nous aimerions évidemment trouver d'autres variables de contrôle qui fonctionnent bien. Par exemple, nous pourrions essayer de remplacer le nombre d'arrivées par une mesure de dispersion (*burstyness* en Anglais) telle que la variance du nombre d'arrivées pendant les périodes. Dans le cas où le modèle ne diffère pas trop de celui imposé par une approximation telle qu'Erlang C ou Erlang A, une autre possibilité consiste à utiliser comme variable de contrôle le résultat de cette approximation. Puisqu'il faut alors simuler les deux systèmes, la réduction de variance doit être importante pour compenser le coût de calcul plus élevé.

Parfois, une variable aléatoire Y peut être décomposée sous la forme

$$Y = \sum_{p=1}^{P} w_p Y_p.$$

Par exemple, Y_p peut être le nombre de contacts servis pendant une période p de la journée et Y, le nombre total de contacts servis. Nous pouvons alors appliquer une variable de contrôle A sur Y globalement ou encore appliquer une variable A_p sur chaque composante Y_p. Les variables A et A_p peuvent par exemple correspondre au nombre total d'arrivées et au nombre d'arrivées durant la période p, respectivement. Il est possible que la corrélation entre Y_p et A_p, pour $p = 1, \ldots, P$, soit supérieure à la corrélation entre Y et A si bien qu'appliquer P variables de contrôle au lieu d'une seule pourrait intuitivement réduire davantage la variance. Par contre, minimiser la variance sur chaque composante séparément, en trouvant une constante β_p optimale pour chaque valeur Y_p, ne minimise pas nécessairement la variance sur Y en raison de la corrélation entre les valeurs de Y_p. Nous aimerions traiter ce problème en estimant les constantes simultanément pour toutes les périodes. Le vecteur $\beta_1, \ldots, \beta_P$ pourrait être estimé par recherche locale à partir du vecteur obtenu en trouvant la constante optimale pour chaque composante séparément ou en considérant Y comme une fonction linéaire de plusieurs moyennes sur laquelle nous appliquerions P variables de contrôle. Si cette technique ne fonctionne pas quand P est grand, nous pouvons approximer le vecteur $\beta_1, \ldots, \beta_P$ en négligeant les corrélations d'ordre élevé. Nous voudrions généraliser cette idée à plusieurs variables de contrôle et à une fonction de plusieurs moyennes.

7.2 Application de la stratification aux centres de contacts

Nous voulons appliquer la stratification, décrite dans un contexte général à la section 6.7, aux centres de contacts. Supposons pour cela que les arrivées des contacts suivent un processus de Poisson doublement stochastique dont le taux d'arrivée au temps t est $B\lambda(t)$, où B est une variable aléatoire de moyenne 1 qui représente le facteur d'acha-

landage de la journée. Ce genre de processus d'arrivées est présenté et justifié dans [52] et [5]. Nous pouvons stratifier sur B puisqu'il contribue beaucoup à la variance des mesures de performance. Nous pouvons utiliser l'allocation proportionnelle ou optimale, car il est facile de fixer B avant de démarrer la simulation. Pour stratifier sur le facteur B qui est une variable aléatoire continue, nous utilisons un vecteur $\mathbf{U} = (U)$ de dimension 1 et fixons $B = F_{\mathrm{B}}^{-1}(U)$ et $p_s = 1/m$. Ici, $F_{\mathrm{B}}^{-1}(U)$ est la fonction de répartition inverse de B.

Dans l'article [36] que nous avons soumis en mars 2006 et qui est une contribution pour ce projet de doctorat, nous avons appliqué cette technique pour estimer $\mathbb{E}[S_{\mathrm{G}}(s,t_1,t_2) + L_{\mathrm{G}}(s,t_1,t_2)]$, le nombre total de clients ayant attendu moins de s unités de temps avant d'être servis ou d'abandonner durant une journée. Pour aller plus loin, nous avons généralisé la stratification pour des fonctions de plusieurs moyennes à la section 6.7, ce qui est une contribution additionnelle pour ce projet. Nous voudrions tester empiriquement l'estimateur $g(\bar{\mathbf{X}}_{\mathrm{strat},n,m})$ que nous avons proposé dans le cas du niveau de service (2.1) et du temps de réponse (2.4), qui sont des rapports de deux espérances.

Nous allons aussi tenter de construire d'autres exemples pour lesquels nous pouvons stratifier sur une autre variable que B. Par exemple, le nombre d'agents pourrait être une variable aléatoire (voir section 4.1) fixée pour la journée sur laquelle nous pourrions stratifier. Les variables de stratification choisies doivent réduire la variance efficacement tout en correspondant à des concepts s'appliquant aux centres de contacts réels.

7.3 Combinaison des variables de contrôle avec la stratification

Combiner la stratification et les variables de contrôle n'est pas simple, car les constantes pour les variables de contrôle dépendent de la variable sur laquelle nous stratifions. Pour estimer une seule espérance avec une seule variable de contrôle, la méthode de combinaison consiste à remplacer chaque observation $j = 1,\ldots,n_s$ de chaque strate $s = 1,\ldots,m$, notée $X_{s,j}$, par $X_{\mathrm{c},s,j} = X_{s,j} - b_{s,j}(A_{s,j} - e_{s,j})$, où $A_{s,j}$ est la valeur de la variable de

contrôle correspondant à l'observation $X_{s,j}$. La valeur de $e_{s,j}$ peut être une constante globale $a = \mathbb{E}[A]$, une constante spécifique à la strate $a_s = \mathbb{E}[A_{s,j}]$ ou encore une constante spécifique à l'observation $a(U_{s,j}) = \mathbb{E}[A \mid B = F_{\mathrm{B}}^{-1}(U_{s,j})]$. Quant à $b_{s,j}$, nous pouvons employer une constante globale β, une constante par strate β_s ou encore une constante $\beta(U_{s,j})$ dépendant de $U_{s,j}$, la valeur utilisée pour générer B.

Nous avons découvert qu'utiliser une constante différente pour chaque strate peut produire de meilleurs résultats que fixer une constante pour chaque valeur de B. Ce résultat est surprenant, car la seconde méthode utilisant davantage d'informations que la première devrait intuitivement donner les meilleurs résultats. Il peut s'expliquer par le fait que dans la décomposition $\mathrm{Var}[X \mid S = s] = \mathbb{E}[\mathrm{Var}[X \mid B]] + \mathrm{Var}[\mathbb{E}[X \mid B]]$, la seconde méthode ne réduit que l'espérance de la variance tandis que la première réduit les deux termes de la décomposition. Ce problème est examiné dans [37] que nous avons soumis en mars 2007 et qui représente une contribution à l'avancement de ce projet.

Nous aimerions trouver un exemple réaliste pour lequel il est mieux de faire varier la constante β en fonction de la variable de stratification continue que fixer une constante par strate. Nous allons aussi construire d'autres exemples combinant stratification et variables de contrôle utilisant d'autres variables que le nombre d'arrivées A ainsi que des fonctions de plusieurs moyennes. Cela va peut-être mettre au jour de nouveaux problèmes.

7.4 Preuve que les variables aléatoires communes réduisent la variance d'une différence dans des contextes spécifiques

Comme nous l'avons vu dans la section 6.2, il est possible d'utiliser les variables aléatoires communes pour estimer des dérivées. Cela revient à estimer des différences avec un δ très petit. Dans ces cas-là, nous pouvons parfois démontrer que la variance est réduite en théorie par un facteur dépendant de δ. En particulier, dans l'exemple utilisé dans [36] et décrit à la section 5.2, nous pouvons multiplier les temps de service par

$1-\delta$, où δ est très petit. De cette façon, la forme de la répartition des temps de service demeure constante, mais la moyenne dépend de δ. Cela permet d'analyser la sensibilité du système à une variation de temps de service pour par exemple permetter à des gestionnaires de centres de contacts de tester si cela vaut la peine de former les agents pour les rendre plus efficaces et ainsi diminuer les temps de service.

Dans [36], nous avons démontré que pour cet exemple, la variance de la différence du nombre de contacts ayant attendu moins de s unités de temps était dans $\mathscr{O}(\delta^{1-\varepsilon})$ pour $\varepsilon > 0$. Pour ce même modèle, nous avons aussi réussi à montrer que la variance de la différence du nombre d'abandons était elle aussi dans $\mathscr{O}(\delta^{1-\varepsilon})$. Ceci représente déjà une contribution dans le cadre de notre projet de thèse.

Pour aller plus loin, nous aimerions démontrer que cette convergence a lieu pour d'autres mesures de performance telles que le temps d'attente, le taux d'occupation, etc. Nous aimerions également prouver cette convergence si les mesures de performance variaient en fonction d'autres paramètres tels que le temps de patience ou le paramètre α_0 du facteur d'achalandage qui suit la loi gamma(α_0, α_0) pour cet exemple. Nous aimerions construire ces preuves de convergence avec le moins d'hypothèses possible au sujet du centre de contacts et idéalement montrer que la variance de la différence est dans $\mathscr{O}(\delta^2)$ dans certains cas puisque cela nous assurerait que la variance sur la dérivée ne dépend pas de δ.

Pour démonter cette convergence, il existe des théorèmes qui s'appliquent pour $\mathbf{X} = (X)$ et qui sont présentés dans [34]. Soit $f(\theta, \mathbf{U})$ la valeur de X pour θ et $\mathbf{U}$ fixés et $\Delta = f(\theta+\delta, \mathbf{U}_2) - f(\theta, \mathbf{U}_1)$. D'abord, si $f(\theta, \mathbf{U})$ est continue en θ et non dérivable (par rapport à θ) seulement pour un nombre fini de valeurs de θ et si la dérivée est bornée par une variable aléatoire dont le second moment est fini [41], alors $\mathrm{Var}[\Delta] = \mathscr{O}(\delta^2)$. Sans cette condition, démontrer la borne devient plus complexe. Selon le théorème 6.5 présenté dans [34], il faut d'abord borner la probabilité que Δ^2 excède une variable aléatoire qui ne dépend pas de θ. Ensuite, l'inégalité de Holder permet d'aboutir à une borne si certains moments d'ordre supérieur à 2 de Δ sont bornés. Une borne peut aussi être

démontrée lorsque Δ est borné avec probabilité 1.

La plupart des fonctions qui nous intéressent dans les centres de contacts sont discontinues, car elles résultent de sommes de fonctions indicatrices. Par exemple, dans le cas du niveau de service pour un centre de contacts, un seul contact supplémentaire servi après avoir attendu moins de s unités de temps peut faire varier la proportion d'un facteur important, surtout si le nombre d'arrivées est petit. Cela empêche l'application du théorème énoncé dans [41] tandis qu'appliquer le théorème 6.5 de [34] exige de borner des probabilités, ce qui n'est pas toujours facile. Tenter de prouver la réduction de variance pour des modèles de centres de contacts pourrait nous permettre de développer des corollaires à partir du théorème 6.5 qui seraient plus faciles à appliquer.

7.5 Application de Monte Carlo conditionnel à l'estimation de dérivées

Estimer la dérivée d'une mesure de performance par rapport à un paramètre du modèle est souvent difficile dans le cas d'un centre de contacts, car comme nous l'avons vu à la section précédente, plusieurs mesures sont discontinues. Dans ce cas, même avec les variables aléatoires communes, l'estimateur n'est pas toujours stable. Par contre, dans certains modèles, il est possible d'appliquer Monte Carlo conditionnel (voir section 6.4) pour rendre la fonction qui nous intéresse continue, ce qui améliore la précision.

Par exemple, dans un centre de contacts, au moment où un contact sort du système, nous connaissons son temps d'attente et savons s'il excède ou non le seuil s pour le niveau de service. Une façon d'appliquer la technique Monte Carlo conditionnel serait de déterminer, à l'arrivée du contact, sa probabilité d'être servi en moins de s unités de temps.

Nous allons d'abord tenter de calculer ou au moins d'estimer cette probabilité. Nous aimerions aussi trouver une expression pour la probabilité d'abandon pour un contact nouvellement arrivé. Nous allons ensuite combiner cette application de la technique Monte Carlo conditionnel avec les variables aléatoires communes pour estimer une dé-

rivée par rapport à divers paramètres tels que le temps de service, les taux d'arrivée, etc.

7.6 Application de l'échantillonnage stratégique pour estimer la probabilité que le niveau de service soit inférieur à un seuil

Bell Canada doit obtenir un niveau de service mensuel de 80%. Pendant le mois, le niveau de service quotidien ou horaire peut être sous le seuil ou au-dessus de ce dernier, mais si le niveau mensuel est au-dessous du seuil à la fin du mois, l'entreprise doit payer une importante amende.

L'échantillonnage stratégique (voir section 6.5) est souvent utilisé pour estimer la probabilité d'événements rares. Nous pourrions alors l'appliquer pour estimer la probabilité que le niveau de service mensuel soit inférieur à 80%. Cela implique d'abord et avant tout de simuler un mois entier. Cela ne pose aucun problème technique au niveau du logiciel, mais il est important de disposer de données réalistes pour tout le mois simulé. De plus, sur une période aussi longue, il est primordial de simuler les recours qu'ont les gestionnaires en cas de baisse du niveau de service.

Si nous parvenons à modéliser un mois de façon réaliste, il nous reste encore à trouver un changement de densité permettant de réduire la variance. Ce changement pourrait par exemple augmenter la probabilité que le niveau de service soit inférieur au seuil. Il reste ensuite à démontrer que ce changement réduit bel et bien la variance.

Si le modèle traité correspond à une chaîne de Markov, ce qui est le cas pour une simulation à événements discrets, il existe un changement de densité permettant d'aboutir à une variance nulle. Par contre, la nouvelle densité est très complexe. L'article [19] expose une technique que nous pourrions tenter d'appliquer pour approximer cette fonction de densité idéale. Nous pourrions également nous inspirer des techniques de preuve de cet article pour démontrer que notre nouvelle densité est efficace pour réduire la variance.

7.7 Réduction de la dimension effective du problème pour quasi-Monte Carlo

Comme nous l'avons mentionné à la section 6.6, les méthodes quasi-Monte Carlo randomisées fonctionnent bien en faible dimension, mais elles deviennent moins efficaces si la dimension est élevée. Pour simuler un centre de contacts, il nous faut malheureusement beaucoup de variables aléatoires, d'où une dimension très élevée.

La première étape pour réduire la dimension est de n'appliquer la méthode que sur une fraction des variables aléatoires utilisées pour la simulation, les autres variables étant générées de façon aléatoire comme avec une simulation classique. Nous devons dans ce cas déterminer sur quelles variables la méthode sera appliquée, notre premier choix étant le nombre d'arrivées. En effet, lorsque ce nombre est fixé, il reste peu de variabilité dans le modèle. Avec P périodes principales, la dimension du problème est alors P. Nous allons tenter de réduire cette dimension encore plus en générant d'abord le nombre d'arrivées pour toute la journée. Conditionnellement à ce nombre, nous générerions le nombre d'arrivées pendant les $P/2$ premières périodes et les $P/2$ périodes suivantes. Ensuite, nous pourrions obtenir le nombre d'arrivées dans les $P/4$ premières périodes et ainsi de suite.

7.8 Implantation efficace des techniques de scission

Durant l'été 2007, nous avons ajouté les éléments nécessaires pour effectuer des réplications en parallèle à SSJ, la bibliothèque de simulation utilisée par ContactCenters. Cela va nous permettre d'implanter efficacement les méthodes de scission présentées à la section 6.8. Dans le cas de la scission visant à réduire la variance dans les périodes achalandées (voir section 6.8.1), nous allons tester la méthode telle que présentée, avec une valeur assez élevée de n pour bien approximer les distributions G_p et ainsi réduire l'effet de la dépendance des réplications parallèles entre elles.

Nous allons ensuite tester la validité de l'approximation (6.19) de $\mathrm{Var}[\hat{Y}]$, qui néglige la dépendance entre les réplications parallèles, en la comparant avec le résultat si

nous répétons l'expérience plusieurs fois. Nous allons également tenter d'améliorer cette approximation en utilisant la décomposition $\mathrm{Var}[Y_p] = \mathbb{E}[\mathrm{Var}[Y_p \mid G_p]] + \mathrm{Var}[\mathbb{E}[Y_p \mid G_p]]$.

Pour ce qui est de l'ajustement du nombre de réplications par période, nous avons trouvé une méthode qui permet d'obtenir une approximation ne tenant pas compte de la corrélation entre les périodes. Mais les vecteurs $\mathbf{n}$ produits par cette méthode ne sont pas optimaux si la corrélation entre les périodes est élevée. Nous aimerions développer des variantes de cette technique qui tiendraient compte, peut-être partiellement, de cette corrélation. Nous allons tester ces nouvelles techniques et les comparer avec celle que nous avons proposée dans ce document. Dans certains modèles, nous pourrions également estimer les n_p sans expérience pilote, ce qui permettrait de réduire le coût de simulation. Par exemple, pour estimer le niveau de service dans un centre de contacts, nous pouvons utiliser une approximation telle que Erlang A. Le nombre de contacts servis ayant attendu moins de s unités de temps peut ensuite être considéré comme une variable aléatoire binomiale dont la variance est connue. Cela nous donne une (grossière) estimation de l'écart-type qui peut être utilisée pour trouver des valeurs de n_p. Nous allons tester de telles heuristiques et comparer les solutions obtenues avec les solutions optimales.

La scission adaptative est une alternative intéressante qui éviterait d'avoir à fixer les valeurs de n_p. Avec cette méthode, une scission a lieu lorsque le système devient achalandé, par exemple si la taille de la file dépasse un certain seuil, et une réplication parallèle peut être tuée si le système simulé devient presque vide. Les points de scission et de fusion ne correspondent pas nécessairement, dans ce cas, avec le début des périodes.

Pour ce qui est de la scission pour l'estimation des sous-gradients, le problème principal réside dans la fusion des réplications filles avec la réplication mère. Il faut, pour déterminer si une fusion est nécessaire, comparer l'état du système avec plusieurs états sauvegardés. Ceci peut être accéléré en utilisant une fonction de hachage qui calcule un nombre à partir d'un état, mais le calcul répété de cette fonction risque de ralentir le système. Nous allons d'abord tester s'il est possible de nous en sortir sans comparer l'état. Pour ce faire, une réplication fille créée à la période p pourrait être tuée arbitrairement

seulement à la fin de la période $p+q$, avec $q > 1$, où l'impact d'un changement d'un paramètre concernant la période p seulement serait minime.

Nous allons également tenter d'utiliser la fusion en ajustant la fréquence à laquelle les comparaisons d'états se produisent. Plus la fréquence de comparaison est élevée, plus le coût en temps de calcul pour les comparaisons est élevé mais plus les fusions se produisent tôt, épargnant des calculs inutiles. Par contre, plus la fréquence de comparaison est basse, plus la fusion risque de se produire de façon tardive, entraînant davantage de calculs inutiles. Nous voudrions également améliorer la technique pour permettre à des réplications filles de se scinder elles aussi et de fusionner entre elles.

CHAPITRE 8

CONCLUSION

Les centres de contacts contemporains sont des systèmes très complexes si bien que seule la simulation permet de les modéliser avec précision. Les résultats obtenus en simulant de tels centres sont souvent contre-intuitifs et très intéressants. Un logiciel flexible pour simuler de tels centres est déjà une contribution importante à l'avancement des connaissances. Il permettra en effet d'effectuer de nombreuses expérimentations pour étudier divers scénarios dans les centres de contacts et constitue une base fondamentale pour tout logiciel d'optimisation de centres de contacts utilisant la simulation.

Un simulateur générique flexible permettra d'expérimenter de nouveaux éléments avec peu de programmation. L'interaction avec d'autres logiciels facilitera l'entrée des données et l'analyse des résultats.

Les extensions que nous apporterons à notre modèle de centre de contacts rendront ContactCenters plus utile puisqu'il simulera un modèle plus réaliste. Cela nous permettra également de découvrir des aspects de modélisation qui n'ont jamais été traités auparavant par la simulation et les formules analytiques.

Grâce à l'analyse de sensibilité détaillée que nous pourrons effectuer à l'aide de notre logiciel, nous pourrons avoir une meilleure idée des aspects sur lesquels nous devons nous concentrer pendant la modélisation. Les résultats de cette analyse et leur interprétation nous permettront de dégager des observations sur le comportement de centres de contacts.

Pour améliorer l'efficacité de ContactCenters, nous sommes appelés à développer des techniques de réduction de la variance. Nous espérons que plusieurs des techniques que nous développerons au cours de ce projet pourront être réutilisées pour d'autres systèmes que les centres de contacts.

BIBLIOGRAPHIE

[1] Altova. XMLSpy — XML editor for modeling, editing, transforming, & debugging XML technologies, 2007. Voir `http://www.altova.com/products/xmlspy/xml_editor.html`.

[2] AspectJ. The AspectJ project at Eclipse.org, 2006. Disponible sur `http://www.aspectj.org`.

[3] J. Atlason, M. A. Epelman et S. G. Henderson. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research*, 127:333–358, 2004.

[4] A. N. Avramidis, W. Chan et P. L'Ecuyer. Staffing multi-skill call centers via search methods and a performance approximation. Soumis, révisé en mai 2007, 2006.

[5] A. N. Avramidis, A. Deslauriers et P. L'Ecuyer. Modeling daily arrivals to a telephone call center. *Management Science*, 50(7):896–908, 2004.

[6] A. N. Avramidis, M. Gendreau, P. L'Ecuyer et O. Pisacane. Simulation-based optimization of agent scheduling in multiskill call centers. Dans *Proceedings of the 2007 Industrial Simulation Conference*. Eurosis, 2007.

[7] A. N. Avramidis et P. L'Ecuyer. Modeling and simulation of call centers. Dans *Proceedings of the 2005 Winter Simulation Conference*, pages 144–152. IEEE Press, 2005.

[8] A. N. Avramidis et J. R. Wilson. Integrated variance reduction strategies for simulation. *Operations Research*, 44:327–346, 1996.

[9] T. E. Booth et S. P. Pederson. Unbiased combinations of nonanalog Monte Carlo techniques and fair games. *Nuclear Science and Engineering*, 110:254–261, 1992.

[10] P. Bratley, B. L. Fox et L. E. Schrage. *A Guide to Simulation*. Springer-Verlag, New York, seconde édition, 1987.

[11] E. Buist. Conception et implantation d'une bibliothèque pour la simulation de centres de contacts. Mémoire de maîtrise, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, août 2005.

[12] E. Buist et P. L'Ecuyer. *ContactCenters : A Java Library for Simulating Contact Centers*, 2005. Manuel de l'utilisateur disponible sur `http://www.ericbuist.com/contactcenters`.

[13] E. Buist et P. L'Ecuyer. A Java library for simulating contact centers. Dans *Proceedings of the 2005 Winter Simulation Conference*, pages 556–565. IEEE Press, 2005.

[14] M. T. Ceẓik et P. L'Ecuyer. Staffing multiskill call centers via linear programming and simulation. *Management Science*, 53, 2007. À paraître.

[15] J. Clark. XSL transformations (XSLT), novembre 1999. Voir `http://www.w3.org/TR/xslt`.

[16] J. Clark et S. DeRose. XML path language (XPath), novembre 1999. Voir `http://www.w3.org/TR/xpath`.

[17] W. G. Cochran. *Sampling Techniques*. John Wiley and Sons, New York, seconde édition, 1977.

[18] CRTC. Final standards for quality of service indicators for use in telephone company regulation and other related matters, 2000. Canadian Radio-Television and Telecommunications Commission, Décision CRTC 2000-24. Voir `http://www.crtc.gc.ca/archive/ENG/Decisions/2000/DT2000-24.htm`.

[19] P.-T. de Boer, P. L'Ecuyer, G. Rubino et B. Tuffin. Estimating the probability of a rare event over a finite horizon. Dans *Proceedings of the 2007 Winter Simulation Conference*. IEEE Press, 2007. À paraître.

[20] V. Demers, P. L'Ecuyer et B. Tuffin. A combination of randomized quasi-monte carlo with splitting for rare-event simulation. Dans *Proceedings of the 2005 European Simulation and Modeling Conference*, pages 25–32, Ghent, Belgium, 2005. EUROSIS.

[21] A. Deslauriers. Modélisation et simulation d'un centre d'appels téléphoniques dans un environnement mixte. Mémoire de maîtrise, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, février 2003.

[22] G. S. Fishman. *Monte Carlo : Concepts, Algorithms, and Applications*. Springer Series in Operations Research. Springer-Verlag, New York, 1996.

[23] E. Gamma, R. Helm, R. Johnson et J. Vlissides. *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Mass., seconde édition, 1998.

[24] N. Gans, G. Koole et A. Mandelbaum. Telephone call centers : Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5: 79–141, 2003.

[25] P. W. Glynn. Efficiency improvement techniques. *Annals of Operations Research*, 53:175–197, 1994.

[26] P. W. Glynn et R. Szechtman. Some new perspectives on the method of control variates. Dans K.-T. Fang, F. J. Hickernell et H. Niederreiter, éditeurs, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 27–49, Berlin, 2002. Springer-Verlag.

[27] P. W. Glynn et W. Whitt. Indirect estimation via $L = \lambda w$. *Operations Research*, 37:82–103, 1989.

[28] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Videira Lopes, J.-M. Loingtier et J. Irwin. Aspect-oriented programming. Dans *Proceedings of ECCOP '97*, juin 1997.

[29] S. S. Lavenberg et P. D. Welch. A perspective on the use of control variables to increase the efficiency of Monte Carlo simulations. *Management Science*, 27:322–335, 1981.

[30] A. M. Law et W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, troisième édition, 2000.

[31] P. L'Ecuyer. Polynomial integration lattices. Dans H. Niederreiter, éditeur, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 73–98, Berlin, 2004. Springer-Verlag.

[32] P. L'Ecuyer. *SSJ : A Java Library for Stochastic Simulation*, 2004. Manual de l'utilisateur, disponible sur `http://www.iro.umontreal.ca/~lecuyer`.

[33] P. L'Ecuyer. Modeling and optimization problems in contact centers. Dans *Proceedings of the Third International Conference on Quantitative Evaluation of Systems (QEST'2006)*, pages 145–154, University of California, Riversdale, 2006. IEEE Computing Society.

[34] P. L'Ecuyer. *Stochastic Simulation*. 2006. Notes pour un cours gradué en simulation.

[35] P. L'Ecuyer et E. Buist. Simulation in Java with SSJ. Dans *Proceedings of the 2005 Winter Simulation Conference*, pages 611–620. IEEE Press, 2005.

[36] P. L'Ecuyer et E. Buist. Variance reduction in the simulation of call centers. Dans *Proceedings of the 2006 Winter Simulation Conference*, pages 604–613. IEEE Press, 2006.

[37] P. L'Ecuyer et E. Buist. On the interaction between stratification and control variates, with illustrations in a call center simulation. Soumis, 2007.

[38] P. L'Ecuyer, V. Demers et B. Tuffin. Splitting for rare-event simulation. Dans *Proceedings of the 2006 Winter Simulation Conference*, pages 137–148. IEEE Press, 2006.

[39] P. L'Ecuyer et C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.

[40] P. L'Ecuyer, L. Meliani et J. Vaucher. SSJ : A framework for stochastic simulation in Java. Dans E. Yücesan, C.-H. Chen, J. L. Snowdon et J. M. Charnes, éditeurs, *Proceedings of the 2002 Winter Simulation Conference*, pages 234–242, 2002.

[41] P. L'Ecuyer et G. Perron. On the convergence rates of IPA and FDC derivative estimators. *Operations Research*, 42(4):643–656, 1994.

[42] P. L'Ecuyer et F. Vázquez-Abad. Functional estimation with respect to a threshold parameter via dynamic split-and-merge. *Discrete Event Dynamic Systems : Theory and Applications*, 7(1):69–92, 1997.

[43] SyncRO Soft Ltd. <oXygen/> XML editor & XSLT debugger, 2007. Voir `http://www.oxygenxml.com`.

[44] V. Mehrotra et J. Fama. Call center simulation modeling : Methods, challenges, and opportunities. Dans *Proceedings of the 2003 Winter Simulation Conference*, pages 135–143. IEEE Press, 2003.

[45] B. L. Nelson. Control-variate remedies. *Operations Research*, 38:974–992, 1990.

[46] NovaSim. ccProphet — simulate your call center's performance, 2003. Voir `http://www.novasim.com/CCProphet/`.

[47] Rockwell Automation, Inc. Arena simulation, 2005. Voir `http://www.arenasimulation.com`.

[48] R. J. Serfling. *Approximation Theorems for Mathematical Statistics*. Wiley, New York, 1980.

[49] C. M. Sperberg-McQueen et H. Thompson. W3C XML Schema, avril 2000. Voir `http://www.w3.org/XML/Schema`.

[50] Sun Microsystems, Inc. *The Java Architecture for XML Binding (JAXB) 2.1*, décembre 2006. Disponible sur `http://jcp.org/en/jsr/detail?id=222`.

[51] R. B. Wallace et W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management*, 7(4):276–294, 2005.

[52] W. Whitt. Dynamic staffing in a telephone call center aiming to immediately answer all calls. *Operations Research Letters*, 24:205–212, 1999.

[53] Wikipedia. Virtual queue, avril 2007. Voir `http://en.wikipedia.org/wiki/Virtual_queue`.

[54] F. Yergeau, T. Bray, J. Paoli, C. M. Sperberg-McQueen et E. Maler. *Extensible Markup Language (XML) 1.0*. W3C Recommendation, troisième édition, février 2004. Aussi disponible sur `http://www.w3.org/TR/REC-xml`.