Université de Montréal

Staffing Optimization with Chance Constraints in Call Centers

par
Thuy Anh Ta

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

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en computer science

Décembre, 2013

© Thuy Anh Ta, 2013.

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

Staffing Optimization with Chance Constraints in Call Centers

présenté par:

Thuy Anh Ta

a été évalué par un jury composé des personnes suivantes:

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

Mémoire accepté le: . . . . . .

RÉSUMÉ

Les centres d'appels sont des éléments clés de presque n'importe quelle grande organisation. Le problème de gestion du travail a reçu beaucoup d'attention dans la littérature. Une formulation typique se base sur des mesures de performance sur un horizon infini, et le problème d'affectation d'agents est habituellement résolu en combinant des méthodes d'optimisatione et de simulation.

Dans cette thèse, nous considérons un problème d'affection d'agents pour des centres d'appels soumis à des contraintes en probabilité. Nous introduisons une formulation qui exige que les contraintes de qualité de service (QoS) soient satisfaites avec une forte probabilité, et définissons une approximation de ce problème par moyenne échantionnale dans un cadre de compétences multiples. Nous établissons la convergence de la solution du problème approximatif vers celle du problème initial quand la taille de l'échantillon croît. Pour le cas particulier où tous les agents ont toutes les compétences (un seul groupe d'agents), nous concevons trois méthodes d'optimisation basées sur la simulation pour le problème de moyenne échantillonnale. Étant donné un niveau initial de personnel, nous augmentons le nombre d'agents pour les périodes où les contraintes sont violées, et nous diminuons le nombre d'agents pour les périodes telles que les contraintes soient toujours satisfaites après cette réduction. Des expériences numériques sont menées sur plusieurs modèles de centre d'appels à faible occupation, au cours desquelles les algorithmes donnent de bonnes solutions, i.e. la plupart des contraintes en probabilité sont satisfaites, et nous ne pouvons pas réduire le personnel dans une période donnée sont introduire de violation de contraintes. Un avantage de ces algorithmes, par rapport à d'autres méthodes, est la facilité d'implémentation.

Mots-clés : centre d'appel, affectation des agents, contraintes en probability, optimisation, simulation, niveau de service, temps d'attente moyen, Erlang C.

ABSTRACT

Call centers are key components of almost any large organization. The problem of labor management has received a great deal of attention in the literature. A typical formulation of the staffing problem is in terms of infinite-horizon performance measures. The method of combining simulation and optimization is used to solve this staffing problem.

In this thesis, we consider a problem of staffing call centers with respect to chance constraints. We introduce chance-constrained formulations of the scheduling problem which requires that the quality of service (QoS) constraints are met with high probability. We define a sample average approximation of this problem in a multiskill setting. We prove the convergence of the optimal solution of the sample-average problem to that of the original problem when the sample size increases. For the special case where we consider the staffing problem and all agents have all skills (a single group of agents), we design three simulation-based optimization methods for the sample problem. Given a starting solution, we increase the staffings in periods where the constraints are violated, and decrease the number of agents in several periods where decrease is acceptable, as much as possible, provided that the constraints are still satisfied. For the call center models in our numerical experiment, these algorithms give good solutions, i.e., most constraints are satisfied, and we cannot decrease any agent in any period to obtain better results. One advantage of these algorithms, compared with other methods, that they are very easy to implement.

Keywords: Call center, staffing, chance constraints, optimization, simulation, service level, average waiting time, Erlang C.

CONTENTS

RÉSUMÉ . . . . . **iii**

ABSTRACT . . . . . **iv**

CONTENTS . . . . . **v**

LIST OF TABLES . . . . . **viii**

LIST OF FIGURES . . . . . **ix**

LIST OF ABBREVIATIONS . . . . . **xii**

NOTATION . . . . . **xiii**

ACKNOWLEDGMENTS . . . . . **xiv**

CHAPTER 1: INTRODUCTION **1**

1.1 Description of call centers 2

1.1.1 Inbound call handling 2

1.1.2 Performance measures 3

1.1.3 Workforce Management 5

1.1.4 Description of emergency call centers 6

1.2 Literature review 7

1.2.1 Modeling a call center 8

1.2.2 Simulation tools 8

1.2.3 Call center staffing problem 9

1.3 Master project 11

1.4 Structure of the thesis 12

CHAPTER 2: MODEL AND PROBLEM FORMULATION **14**

2.1 Erlang C traffic model on call centers 14

2.2 Performance measure 16
2.3 Description of the model 19

CHAPTER 3: SAMPLE AVERAGE APPROXIMATION OF THE CHANCE-CONSTRAINED STAFFING PROBLEM 26

3.1 Almost Sure Convergence of Optimal Solutions of the Sample Average Approximation Problem 28
3.2 Exponential Rate of Convergence of Optimal Solutions of the Sample Problems 34

CHAPTER 4: SIMULATION METHODS 37

4.1 General idea for simulation algorithms 37
4.2 Simulation algorithm 1: CCS1 39
4.3 Simulation algorithm 2: CCS2 40
4.4 Simulation algorithm 3: CCS3 40
4.5 Analysis of the algorithms 43
4.6 Cutting plane method for chance-constrained staffing problem 46
4.7 Out-of-sample analysis 47

CHAPTER 5: NUMERICAL EXPERIMENTS 48

5.1 Arrival process 48
5.2 An emergency call center 49
5.2.1 Data from an emergency call center 49
5.2.2 A call center with very low occupancy 50
5.2.3 A low occupancy call center 58
5.3 A call center with high occupancy 63
5.3.1 Parameters 63
5.3.2 Analysis staffing levels obtained by Erlang C and method CCS3 64
5.3.3 Analysis of the solutions of our algorithms 64
5.4 A call center with larger arrival rate 67
5.4.1 Parameters 67

5.4.2 Analysis of staffings obtained by Erlang C and CCS3 69
5.4.3 Analysis of the solutions of our algorithms 70
5.5 Summary 72

CHAPTER 6: CONCLUSIONS AND FURTHER RESEARCH PERSPECTIVES 75
6.1 Conclusions 75
6.2 Further research perspectives 77

BIBLIOGRAPHY 78

LIST OF TABLES

5.I *Violated constraints for out-of-sample simulations of the ten models.* 56

5.II *Mean and standard deviation of the staffing costs with the sample size 1000.* 57

5.III *The violated constraints for the out-of-sample simulation of the ten models with low occupancy.* 62

5.IV *The constraints which are not satisfied for the out-of-sample simulation of the eight models.* 72

LIST OF FIGURES

5.1 *Total costs of final solutions with the three algorithms for 1000 replications.* 52

5.2 *Staffing levels obtained from Erlang C and algorithm* `CSS3` *for 1000 replications of* `MondayPGB`. 52

5.3 *The distribution of the SL in the whole day of the model* `MondayPGNR` *with the staffing level obtained by Erlang C.* 53

5.4 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by Erlang C.* 54

5.5 *Proportion of the days where the AWT constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by Erlang C.* 55

5.6 *The distribution of the service level in the whole day of the model* `MondayPGNR` *with the staffing level obtained by CCS3.* 57

5.7 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by CCS3.* 58

5.8 *Proportion of the days where the AWT constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by CCS3.* 59

5.9 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by increasing one agent in period 34 from the staffing level obtained by CCS3.* 60

5.10 *Proportion of the days where the SL constraint was satisfied, for each period, over 10000 simulated days, for the* `MondayPGBNR` *model, with the staffing level obtained by decreasing one agent in period 12 from the staffing level obtained by CCS3.* 61

5.11 *Staffing levels obtained by Erlang C and CCS3 of the* `MondayPGB`* *model.* 62

5.12 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGB`* *model with the staffing level obtained by Erlang C.* 63
5.13 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGB`* *model with the staffing level obtained by CCS3.* 64
5.14 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGNR`* *model, with the staffing level obtained from CCS3.* 65
5.15 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGNR`* *model, with the staffing level obtained by increasing one agent in period 43 from the staffing level obtained by CCS3.* 66
5.16 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGBNR`* *model, with the staffing level obtained by decreasing one agent in period 8 from the staffing level obtained by CCS3.* 67
5.17 *Staffing levels obtained by Erlang C and CCS3 of the* `MondayPGB`** *model.* 68
5.18 *Staffing levels obtained from CCS3 before and after adding stage* `Correction` *of* `MondayPGB`**. 68
5.19 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGB`** *model, with the staffing level obtained by CCS3.* 69
5.20 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGB`** *model, with the staffing level obtained by decreasing one agent in period 11 from the staffing level obtained by CCS3.* 70
5.21 *Staffing levels obtained from Erlang C and algorithms for 1000 replications of* `MondayPGB`$_2$. 71
5.22 *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPWCP`$_2$ *model, with the staffing level obtained by CCS3.* 73

5.23 *The distribution of the service level in the whole day of the model* $\texttt{MondayPWCP}_2$ *with the staffing level obtained by CCS3.* 73

5.24 *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* $\texttt{MondayPWCP}_2$ *model, with the staffing level obtained by decreasing one agent in period 46 from the staffing level obtained by CCS3.* 74

LIST OF ABBREVIATIONS

AHT : *Average Handle Time*;

ASA : *Average speed of answer*, average waiting time of calls;

AWT : *Average waiting time*;

CCS1 : *First chance-constrained staffing algorithm*;

CCS2 : *Second chance-constrained staffing algorithm*;

CCS3 : *Third chance-constrained staffing algorithm*;

FCFS : *First come, first served*, routing policy first come, first served;

FIFO : *First in, first out*, a synonym of FCFS;

PG : *Poisson-Gamma*;

PGB : *Poisson-Gamma with busyness factor*;

PGNR : *Poisson-Gamma NORTA rates*;

PWCP : *Piecewise constant Poisson*;

PWCPB : *Piecewise constant Poisson with busyness factor*;

QoS : *Quality of service*;

SL : *Service level*;

SLLN : *Strong law of large number*;

TSF : *Telephone service factor*;

WFM : *Workforce Management*, management of labour;

w.p.1 : *With probability 1*.

NOTATION

c	Vector of size $I \times Q$ that represents the cost of each agent group in each shift;
d	Vector of size $I \times P$ that represents the cost of each agent group in each period;
$\mathbb{E}(\cdot)$	Expectation operator;
I	Number of agent groups;
$i \in \{1,...,I\}$	Agent group i;
K	Number of call types;
$k \in \{1,...,K\}$	Call type k;
$\mathbb{N}$	Set of natural numbers;
$\mathbb{P}(\cdot)$	Probability operator;
P	Number of periods;
$p \in \{1,...,P\}$:	Period p;
Q	Number of shifts;
$q \in \{1,...,Q\}$	Shift q;
$\mathbb{R}$	Set of real numbers;
r	Probabilities which the service level constraints are satisfied;
s	Service level target;
S_i	All types of calls that can be served by agents in group i;
v	Probabilities which the average waiting time constraints are satisfied;
w	Acceptable waiting time in the average waiting time constraints;
x	Vector of size $I \times Q$ representing the number of agents in each group in each shift;
y	Staffing vector of size $I \times P$ that represents the number of agents in each group at each period;
τ	Acceptable waiting times in the service level constraints;
λ	Call arrival rate.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Pierre L'Ecuyer for his support, for the useful comments, remarks and his engagement through the learning process of this master thesis. He has always helped me to see a clearer and bigger picture, listened carefully to my ideas and analysed my idea by the best of his knowledge and belief. I am honoured to be working with such an experienced and intelligent professor. Furthermore, I would like to thank my co-supervisor Fabian Bastin for his valuable comments both in my work and in this thesis, for many interesting discussions. Indeed, he is always enthusiastic whenever I need his help.

I thank my colleges in Simulation and Optimization Lab: Richard Simard, Wyean Chan and Nazim Regnard for the stimulating discussions, for their useful experience as I started my research, and for all the fun we have had in the last two years.

I am grateful to my professor Michel Toulouse, who leaded me to the dream of studying in Montreal, introduced me to my professors Pierre L'Ecuyer and Fabian Bastin, and gave me lots of valuable advices.

Also, I would like to thank my best friend, Tien Mai Anh, who is also my college in my lab. He always encouraged me whenever I met difficulties. Therefore, I am always confident with all his support. I will be grateful forever for his love.

I also want to thank all my Vietnamese friends in Montreal, for their help at the first time I came to this beautiful city, for their valuable experience about the new life and new research environment in Canada. I am grateful to my professors in Department of Computer Science and Operation Research, who gave me precious lessons and paved the way for the completion. I also truly appreciate the excellent service provided by the departmental administrators and departmental secretaries. The list is long so I do not enumerate them here.

Finally, I would like to dedicate this thesis and all of my academic achievements to my parents, my sister and all the people in my family who have been anxiously waiting for its completion.

CHAPTER 1

INTRODUCTION

Telephone call centers are key components of many businesses, and their economic role is growing. They are used to handle customer support, phone orders and sales, marketing, governmental information services (police, ambulances,...), etc. Call centers have become a popular means for companies to communicate with their customers. Some call centers are very essential and it would be really difficult to imagine a government agency, financial institution or 911 emergency services without telephone service.

The call center industry is thus large and rapidly expanding, in terms of both workforce and economic scope. It employs millions of people around the world and is fast growing. In the United States, according to statistics from the Bureau of Labor Statistics [11], agents in customer service rank 7th in the list of the largest occupations in 2010. This agency estimates that there were approximately 2.3 million agents in the U.S. in 2008 with 23% working in the financial and insurance sectors and 15% in the area of administration and support service (Bureau of Labor Statistics [8]). Most of these employees work in call centers, but the data also contains agents who interact directly to clients. The research predicts an increase in the number of jobs by 18% to 2.7 million in 2018. Another report estimated that there were 2.15 million agents in May 2010 with average hourly wages of US $15.76 and median of US $14.64 and an average annual salary of US $32780 (Bureau of Labor Statistics [10]). The annual salary cost of agents is then estimated at US $70.3 billion in 2010 in the United States. For comparison, in May 2007 (before the financial crisis of 2008-09), there were 2.2 million agents and an average annual salary of US $31040 for a total labor cost of US $68.1 billion (Bureau of Labor Statistics [9]). Because call centers typically spend 60% to 70% of their budgets on labor costs (Gans et al. [16]), it is important to optimize the management of labor. Their management is complex and is a major area of application for operations research.

1.1 Description of call centers

Gans et al. [16] and Koole [22] provide a good description of the functioning of a call center and the different stages that a call must pass before being answered by an agent. A call (or contact) represents a communication between a client and a service. A call is also distinguished by its issuer: it can be emitted by a client (*incoming call*) or by an agent (*outgoing call*). These calls are generally classify by type, representing the requested service and its source of origin. An employee who interacts with the customer on the phone is called an *agent* or *representative customer service*. Agents sharing a common set of tasks form a *group of agents*. Each of them requires special skills on the part of the employees: language, technical knowledge of a specific product, etc. A group of agents is called *specialist* if it is assigned to some tasks and *general* in the case of multiple tasks. When the skill-level required to handle calls is low, each employee is trained to handle every type of call, and calls may be handled *first come, first served* (FCFS), also called *first in, first out* (FIFO). Otherwise, if more highly-skilled works are required, each agent may be trained to handle only a subset of the types of calls, a *"skills based routing"* may be used to route calls to appropriate agents. Obviously, a client can be transferred through several staffs before being satisfied. There are various kinds of call centers: *inbound call centers* service ingoing calls, *outbound call centers* handle outgoing calls, a call center that handle these mixed operations is referred to *blend incoming and outgoing center*. In our context, we only consider inbound call centers. A staff is called multi-purpose when an agent can serve several types of calls. Besides the cost of training, the cost of an agent is often determined by the number of tasks assigned to the group.

1.1.1 Inbound call handling

Customers call the centers for various reasons. When a call arrives, a free agent is selected among agent groups. The router uses the type of the call to determine which agents are allowed to serve the call, and how agents are chosen if several agents are free. If a free agent is found, the call is sent to that agent, and the agent is allocated for a

certain *service time*. If no agent is available for a new call, the call is sent to a waiting queue if that does not exceed the total queue capacity. A call entering queue *balks* if it abandons immediately. Other calls having to wait go into a queue where they remain until agents are free to serve them. A queued caller can also become impatient, and *abandon* without service. If the queue is full at the time of an arrival call, the call is *blocked* instead of entering the queue, i.e., the caller receives a busy signal.

1.1.2 Performance measures

Performance measures allow to assess the quality of service and efficiency of a call center. The main purpose of these performance measures is to ensure the call center is meeting its goals and objectives. Among them, *service level* (SL) is one of the most popular. It denotes the percentage of calls that are answered in a defined waiting threshold. The constraint on the SL is most commonly stated as s percent of calls answered in τ seconds or less, where τ is a parameter, and is usually denoted by s/τ. The SL can be measured and controlled separately by time period (hour, day, etc.) and by call type, or in an aggregated way. Many contact center managers simply assume that a target of 80/20 is the industry standard, and therefore use that as their own target. While this may be the most common service level for customer service call centers, the fact is that there is no industry standard for the SL. Other centers such as 911 in Montreal or emergency set their standards to 95/2. Similarly technical support centers often have as target service level waiting times of 3 to 5 minutes for free support (Seyrafiaan [32]). In practice, call centers set their overall target (both percentage of calls and the threshold) in conjunction with their Work Force Management (WFM) division in order to calculate their staff requirements and scheduling. A higher service level means faster service (answering the call) for customers. An important motivation for studying this measure is that for many types of call centers that provide services, in several countries, there are government regulations on the minimal acceptable SL and the call centers may have to pay very large fines when this SL is not met. As we will see in Section 2.2, over a given time period, the SL is a random variable. Therefore, in the optimization point of view, ensuring the target over finite durations can be expressed by chance constraints. One

may prefer to define the SL over a long-term (infinite-horizon) in order to work with expectations only, but this only ensures that the target is met on average.

The definition of SL also encourages us to give priority to calls who waited less than or equal to τ, because serving those who waited more than τ can not improve the measurement. In other words, while the SL indicates the percentage of calls that were answered within the waited threshold, it does not provide any information regarding the remaining calls. For this reason, it is important to look at a measure that represents all the callers, such as the *average waiting time* (AWT) or *average speed of answer* (ASA). The AWT or ASA in a period is the average (or mean) time a customer waited to have a service for this period. For example, in a time interval, if half the calls go into queue and wait for an average of 30 seconds, and the other calls go immediately to an agent, the average waiting time is 15 seconds. Obviously, a lower service level (lower percentage of calls or longer threshold) produces a longer ASA. Combined with the SL, the ASA provides a more complete picture of the flow of the incoming calls. Another measure is the *abandonment ratio*, which is measured by looking at the calls that abandon during the defined time period compared with all calls for that period. Service level and average waiting time are two of quality of service (QoS) measures.

The satisfaction and well-being of agents influence the performance of a call center. A manager often measures the efficient use of call center agents by *the occupancy ratio* of agents. It is the percentage of time an agent is busy on a call or doing after call work compared with available time. It is calculated by dividing time spent to answer calls by total time at the workplace. For a call center with a high volume of calls and a lot of agents, it is often possible to have a high quality of service with an occupancy rate of over 90%. If occupancy is too low, agents are idle. If occupancy is too high, agents are overworked, so they will be less effective, because the agents are exhausted. The art of the workforce management (WFM) process is to create a balance between the SL and the occupancy ratio. Practice shows that for most (though not all) centers, an occupancy ratio of between 75% to 85% is optimal. However, not every call center or agent group can reach that number. Small call centers that wish to deliver an 80/20 service level and have sufficient staffing in place may not be able to achieve occupancies above 70% or

80%. Larger call centers have the opposite problem. Their large group efficiencies may allow them to staff for the same 80/20 service level and have occupancy numbers over 95%. In such cases, these managers have to add extra workers to bring occupancy down to a tolerable level (Reynolds [31]). Many other different performance measurements used to gauge the efficiency and effectiveness of a call center operation are discussed in Reynolds [31].

1.1.3 Workforce Management

Workforce management is an essential part of the operation in any call center. It can be summed up as a series of activities related to forecasting call volumes, and scheduling required and appropriate staff. A complete WFM process is required to create planning documents, call volume forecasts, agents schedules and intra-day adjustments (Seyrafi-aan [33]).

Planning As the name implies, the Planning stage is where it all begins. Using high level forecasting techniques, call centers can come up with their expected annual work-load (work-load is the total time required to handle the arrival calls). From here, the center can calculate the overall staffing requirements, hiring timelines, training requirements and timelines as well as vacation allocation and the total budget.

Forecasting A typical forecast predicts call volumes for any given time intervals in a day (majority of centers use 15 or 30 minutes intervals) for the forecasting period. Forecasting stage is based on the historical call patterns. However, there may be some requirements for final adjustments based on the latest information as well as changes in the environment.

Scheduling After forecasting the number of incoming calls for each interval, the next step is to determine the number of agents required for each interval. But WFM is more than simply determining agents for a day. Managers also schedule lunches, breaks, scheduled trainings and vacations, and deal with the 2-3% of staff that will not show up for their shift.

Intra-day adjustments Even with the best laid plans and calculations, it is necessary to track the operation of the queue (call volumes, service level) and adjust the staffing to ensure that the center is providing the best service level possible, while maintaining a reasonable occupancy rate. An intra-day adjustment team is responsible for tracking and reporting the operational indicators, re-forecasting the daily volumes (usually twice a day), reassigning staff to and from various off-line activities and maintaining the overall target service level.

1.1.4 Description of emergency call centers

Our work focuses on *emergency call centers*. Emergency services call centers are a specialised component of the call center industry. An emergency is any situation where the safety of people or property is at risk and requires immediate assistance. For example, 911 is the emergency telephone number for the North American Numbering Plan. This number is intended for use in emergency circumstances only, and to use it for any other purpose (including non-emergency situations and prank calls) can be a crime. Examples of the 911 emergencies in Canada include: a fire, a crime in progress or a medical emergency. A general view of an emergency call center is a call center where agents are trained to answer calls from emergency situations. In the context of an inbound emergency services call center, the organisation has no control over the arrival rate which depends on natural and human phenomena (Lewis et al. [29]). Although many calls do arrive randomly, some kinds of incidents generate spurts of calls that are related to the same incident and so are not truly random. For example, a visible fire will usually generate many calls from the area. Calls will peak within several minutes and then downgrade as the fire suppression units arrive. Most of the calls within a short period will be related to the same incident, together with random calls also arriving interspersed among the fire calls. Many other kinds of incidents also generate multiple calls, creating a large spurt of call arrivals, and such effects demand more personnel to handle them. However, it is not uncommon to see overlapping spurts, which further stresses the emergency systems. Nobody knows in advance when such bursts can happen, so one cannot put more personnel at the right time. What is possible is to put one or a few more agents most of the

time to cover this possibility. According to Lewis and Herbert [28], the emergency call centers provide support to the community by a high level of service when needed. Their effectiveness is gained by operation through efficient scheduling, rapid response to calls and a high standard of agent capability. Therefore, the service level should be very high and average waiting times are very low. Lafond [23] shows an example where 90% of all 911 calls arriving shall be answered within 10 seconds during the busy hour (the hour each day with the greatest call volume) and 95% of all 911 calls should be answered within 20 seconds. The 911 center in Montreal requires that 95% of all arriving calls shall be answered within 2 seconds. The requirements of high service levels and low average waiting times in emergency call centers imply that the occupancy of agents will be low.

1.2 Literature review

First, we give some definitions of Staffing, Scheduling and Routing Problems of call centers. The goal of these problems is to minimize the operating cost of the center under a set of constraints on certain performance measures such as SL, AWT and so on. One decision to be made is how many agents of each skill group to have in the center as a function of time. In a *staffing* problem, the day is divided into periods (e.g., 30 minutes or one hour each) and one simply decides the number of agents of each group for each period. In a *scheduling* problem, a set of admissible work schedules is first specified, and the decision variables are the number of agents of each skill group in each work schedule. This determines the staffing indirectly, while making sure that it corresponds to a feasible set of work schedules. A yet more restrictive version of the problem is when there is a fixed set of available agents to be scheduled for the day or the week, where each agent has a specific set of skills. Then we have a *scheduling and rostering* problem.

In this thesis, we focus on the staffing problem for inbound call centers. This section provides a survey of the recent literatures and various tools which can help solve this problem.

1.2.1 Modeling a call center

Modelling a call center is difficult because it is common to have only the averages over each period of the day, e.g., a half hour. It is difficult to find the appropriate distributions and dependencies between random variables with such aggregated data.

The *arrival process* of calls is not a homogeneous Poisson process (deterministic rate). However, we often make this assumption, for reasons of mathematical simplicity. More recent studies suggest a double stochastic process, e.g., Poisson-Gamma, if the arrival rate of the Poisson process is a random variable (Avramidis et al. [2], Brown et al. [6], Jongbloed and Koole [19]). Arrival rates often vary depending on the time-of-day and often on the day-of-week. A positive correlation between periods and between days was also observed in several analyses. In this thesis, we consider several arrival processes which are discussed in Oreshkin et al. [30]. We give more details to explain these arrival processes in Section 5.1.

The *service time* of a call is often regarded as a random variable following an exponential distribution. Brown et al. [6] suggests that the lognormal distribution is usually a much better fit.

Several other processes are less studied; data are often unavailable or partially conditional on certain events. The *patience time* determines the time a customer is willing to wait before giving up. It is important to model the patience time distribution correctly because it can have a significant effect on the SL and abandonment ratio. Estimating the patience distribution requires special statistical techniques (Brown et al. [6]).

1.2.2 Simulation tools

Simulation is a very flexible tool which generally requires less advanced knowledge of mathematics that the analytical models. The widening gap between the evolution of reality call centers and the development of analytical models is one of the main reasons for the popularity and the need of simulation tools. However, simulation is a complex program that requires considerable development effort. Based on the collection of statistics, simulation requires significant lead time to reduce the noise and the confidence

interval of the measurements.

For the optimization software discussed in our thesis, we use a simulator built from a Java library for simulating contact centers (Buist and L'Ecuyer [7]), see also `http://www.iro.umontreal.ca/~simardr/contactcenters/index.html`. It is based on the well-supported modern programming language Java and is built over the SSJ simulation library (L'Ecuyer [25], L'Ecuyer and Buist [26]). The much more flexibility than commercial products, the ease of interaction with other Java programs and the fast execution time are some features of this software.

1.2.3 Call center staffing problem

The call center staffing problem has received a great deal of attention in the literature. Staffing in the *single-skill* case (i.e., single call type and single agent type) has received much attention in the call center. literature. It is common to divide the day into several periods during which the staffing is held constant and the arrival rate does not vary much. The system is often assumed to reach steady-state, and steady-state queueing models are used to provide a staffing for each period. The simplest queueing model of a call center is the $M/M/n$ queue, also known as an Erlang C system. This model ignores blocking and customer abandonments. We discussed this system in more detail in Section 2.1. However, this assumption never happens in the real call centers, so it is a crude approximation made in each time period. For better accuracy, the staffing and scheduling should be done using simulation, as in the following articles.

Atlason et al. [1] propose a general methodology, based on the cutting plane method of Kelley [21], to optimize the staffing of agents in a call center with single call type and single skill, under service level constraints. They formulate the constraints in terms of infinite horizon service levels. Their method combines simulation with integer programming and cut generation. First, they relax the staffing problem to a *sample average approximation* (which becomes a deterministic problem). Then, they optimize this sample problem by generating cuts from the violated service level constraints and adding correspond linear constraints. This process terminates until the optimal solution of the relaxed problem is feasible for the original problem.

In the *multiskill* case, the staffing and scheduling problems are more difficult. Staffing a single period in steady-state is already difficult; the Erlang formulas and their approximations (for the SL) no longer apply. Simulation seems to be the only reliable tool to estimate the SL. Ceşik and L'Ecuyer [12] extend the method of Atlason et al. [1] to the multiskill call centers. They also show some difficulties encountered with larger problems, and develop (heuristic) methods to deal with these problems.

In order to solve the *multiskill scheduling problem*, Bhulai et al. [5] propose a two-step approach. The first step determines a staffing of each agent type for each period. In the second step, it solves a linear program to find a set of shifts that cover this staffing, by allowing agents to use only a subset of their skills in certain periods if needed. Bhulai et al. [5] recognize that their two-step approach is generally suboptimal and they illustrate this by examples.

Avramidis et al. [3] propose a simulation-based algorithm for solving the multi-skill scheduling problem, and compare it to the approach of Bhulai et al. [5]. This algorithm extends the method of Ceşik and L'Ecuyer [12], which solves a single period staffing problem.

In typical problem formulations, the constraints with respect to the average performance measures in the long run are considered. Gurvich et al. [17] propose a more appropriate problem formulation, which is to use probabilistic constraints on the (random) values over a given time period. The form is as following: the call center's management chooses a *risk level* δ, and allows the QoS to be violated on at most a fraction δ. They consider the probabilistic constraints on the abandonment ratios, with random (but time-independent) arrival rates, and use a fluid approximation of the abandonment ratios for any realization of the arrival rate. Moreover, they develop a *two-step* method to optimize the staffing (for the staffing problem) under chance constraints. The first step introduces a *Random Static Planning Problem* (RSPP) and discusses how it can be solved using two different methods. The RSPP finds a set of staffing levels that minimize staffing costs subject to the requirement that the staffing levels are sufficient to meet the demand of all types of calls with a probability that is $1-\delta$. The output of the RSPP is a staffing solution and a set of arrival rate vectors which are called the *staffing frontier*. In the

second step, they solve a finite number of staffing problems with known arrival rates – the arrival rates on the optimal staffing frontier. Thus, the most important role of the staffing frontier approach is that it reduces the complex staffing problem with uncertain rates to one of solving multiple problems with predictable rates. In the end, the output is a staffing and routing solution that is feasible to chance constraints and is nearly optimal for large call centers.

1.3 Master project

The management of labour in call centers is a difficult optimization problem. There are lots of studies in this field. Some formulas as well as few algorithms to solve the staffing problem in call centers have been proposed in literature. As discussed above, the authors formulate the constraints in long terms average performance measures over an infinite horizon. This type of formulation stems from historical reasons, but it is not necessarily appropriate for some practical applications. Even if the average is above the target threshold, the service level on a given day is a random variable that may have a large variance, and may take a value much smaller than the target for a significant fraction of the days. Therefore, according to the suggestion of Gurvich et al. [17], in our thesis, we consider the use of probabilistic constraints expressing that the service level targets (per call type, per period, global) on a random day must be satisfied with probability at least $1-\delta$, for a given risk level δ selected by the manager. We are also interested in constraints on the average waiting time. The goal is to design an optimization software to find the optimal allocation while satisfying these constraints, therefore defining a simulation-based optimization problem. We adopt the *"sample-average approximation"* approach for solving this problem, as described in the following. First, we generate the simulation input data for N independent replications of the operations of the call center over the planning horizon. This data includes call arrival times, service times and so forth. When this data is fixed, we can estimate the probabilities in the constraints by the averages computed over the generated realizations. Then, we solve a deterministic optimization problem that chooses staffing levels so as to minimize the staffing cost,

while ensuring that the constraints are satisfied. Then, we study the convergence of the optimal solution of the sample problem to that of the original problem. By using the strong law of large numbers (SLLN), we show that the set of optimal solutions of the sample problem is a subset of the set of optimal solutions for the original problem with probability 1 (w.p.1) as the sample size gets large. Furthermore, we show that the probability of this event approaches 1 exponentially fast when we increase the sample size. We do all of these in a multiskill setting. In this thesis, we focus our attention on solving the sample-average approximation problem for the single agent type call centers, and propose three simulation-based optimization algorithms. Their original idea is increasing the staffing in periods where the constraints are violated, and decreasing the number of agent in each period as much as possible while still satisfying the requirements. These algorithms have some variations to improve their quality in performance. We prove that these methods terminate under some conditions and analyse the quality of the solutions obtained by the algorithms. We also include extensive numerical experiments in several different types of call centers, to define the cases when the methods are robust. In case the methods are not as good as we expect, we find a solution to improve the quality of the results.

1.4 Structure of the thesis

This thesis is divided in six chapters. Considering the interdisciplinary nature of this research, some background material is provided to facilitate the discussion. In Chapter 2, we model the staffing problem by using chance constraints with respect to the service level and the average waiting time. In chapter 3, we define the sample average approximations of the chance-constrained staffing problem in the multiskill setting. This chapter also considers the convergence of optimal solutions of the sample problem to the real optimal staffing level in the original problem. In chapter 4, we propose three simulation-based optimization algorithms to solve the sample average approximations for the special case where all agents have all skills (a single group of agents). Chapter 5 measures the efficiency of each algorithm in several instances. We use the real data from

an emergency call center 911 which has low occupancy. We also test the algorithms in other call centers which have much higher occupancy. Then, we consider the heavy traffic call center models to access the quality of our algorithms.

CHAPTER 2

MODEL AND PROBLEM FORMULATION

A general review on the description and management of a call center was presented in Chapter 1. This chapter introduces the main problem of this thesis in mathematical form. This problem is the management of the workforce for a number of specific groups assuming an infinite number of available agents per group. We define the schedule problem in a multiskill setting in this section. However, we provide the algorithms to solve the staffing problem only for a single-skill call center in the remainder of our thesis.

2.1 Erlang C traffic model on call centers

In this section, we give some definitions for the Erlang C traffic model for a single call type. According to Cooper [14], a queuing model can be defined in terms of three characteristics: the input process, the service mechanism, and the queue discipline. The queuing models which we consider in this section are built under the following assumptions: the customers are assumed to arrive according to a Poisson process with constant rate λ, the service times are assumed to be exponentially distributed with rate μ and independent of each other (as well as everything else in the system); the queue discipline is assumed to be *blocked customers delay*, i.e., when blocked customers wait as long as necessary for service; the waiting calls are handled FIFO; and there are n agents. These queuing models are called the $M/M/n$ queue. The number of waiting positions in the queue is assumed to be infinite. The offered traffic load is defined by $\rho = \frac{\lambda}{\mu}$. Let $C(n,\rho)$ denote the probability that all servers are occupied. According to Cooper [14], the formula to compute $C(n,\rho)$ is:

$$C(n,\rho) = \frac{\frac{\rho^n}{n!(1-\rho/n)}}{\sum_{k=0}^{n-1}\frac{\rho^k}{k!} + \frac{\rho^n}{n!(1-\rho/n)}},\ (0 \le \rho < n). \tag{2.1}$$

This is called the *Erlang delay formula* or *Erlang C formula*. The Erlang C formula

was first published by by A.K. Erlang, a Danish scientist, in 1917. Since an arriving call has to wait if all the servers are busy, the *delay probability* $\mathbb{P}[W > 0]$, where W is the waiting time of a call, is given by (2.1). The SL for a given n is computed from

$$\mathbb{P}[W \leq \tau] = 1 - C(n,\rho)e^{\tau(n\mu-\lambda)} \text{ with } \tau \geq 0.$$

where τ is the acceptable waiting time.

The Erlang C function computes the probability that an arrival call in the Erlang C queueing model will find that all servers are busy. This is the same as the fraction of arrival calls that are delayed (i.e., must wait) before being answered. The service level estimate given by the Erlang C formula is *the average over an infinite time horizon*. The Erlang C formula describes the relationship between the call-arrival rate, the average service time (including the hold or wrap-up time), the service level, and the number of agents. If we know any three of these numbers, we can calculate the unknown factor. Generally, we are calculating the staff number needed to achieve a service level or we are calculating the expected performance (service level) of a number of staffings. The minimum n required to meet a given target of SL s, i.e., $\min_{n\geq 0}\{n : \mathbb{P}\{W \leq \tau\} \geq s\}$, can be obtained by some methods, using the fact that the SL is monotone in n. In our thesis, we use the Erlang C formula and the binary search to find the required number of staffs.

However, when n is very large and the system has high utilization, using the exact Erlang C formula is quite difficult. Halfin and Whitt [18] give an approximation to the Erlang C in this case. They consider a sequence of $M/M/n$ queues for which $n \to \infty$ and $(1-\gamma)\sqrt{n} \to \beta$ for $0 < \beta < 1$, where μ is fixed and $\gamma = \lambda/(n\mu)$ is the associated average system utilization or *occupancy* (also called "traffic intensity"). And they prove that

$$\mathbb{P}[W > 0] \longrightarrow \frac{1}{1+\beta\Phi(\beta)/\phi(\beta)}$$

where Φ and ϕ are the standard normal distribution function and density, respectively. They show that when the offered load ρ is high, and an appropriate number of agents are employed, a system can achieve a high agent utilization and yet deliver a good service level by choosing the number of servers as $n = \rho + \beta\sqrt{\rho} + o(\sqrt{\rho})$. If omitting the

small order term, we call this *square-root safety staffing*. If γ is fixed while λ and μ increase to infinity, then $\mathbb{P}[W > 0]$ converges to 0, i.e., nobody waits in the limit. This is called the *quality-driven* regime. It is appropriate for situations where speed of answer is much more important than the cost of agents, e.g., emergency service. In another case, if the safety staffing $\beta\sqrt{\rho}$ is fixed while both λ and n increase to infinity, then $\mathbb{P}[W > 0]$ converges to 1, so in the limit, everyone waits. This *efficiency-driven* regime is appropriate for call centers where the productivity of agents is much more important than the wait, i.e., an answering email system. Another regime called a *quality and efficiency-driven* (QED), is a regime for which $\mathbb{P}[W > 0]$ is fixed to a constant in $(0,1)$ while λ and n increase to infinity. It is good for a large call center with constraints on the SL and abandonment ratio.

2.2 Performance measure

We describe in more details the performance measures introduced in Section 1.1.2. In reality, the performance measures are calculated from the observed data at the end of a period or day. There is no unique formula or standard for measuring performance. There are often many different formulations to compute these measures. In many optimization problems studied so far, a general approach is to consider the expected performance measures over an infinite time horizon. However, in this context, we would like to consider the distributions of performance measures in a given time interval, not the expected value. We present in this section the performance measures used in this thesis or considered important.

The *service level* (SL) is one of the measures which is most used in industry. This measure is also referred to as *telephone service factor* (TSF). The formula for the SL is not unique, but it can be summed up as *the fraction of calls answered within a given time* τ, where τ is a parameter which is called *acceptable waiting time*. We present only some formulas of service level and distinguish the definitions of service level over a given time period and in a long run. Many other formulas are proposed in Jouini et al. [20]. Let $S(\tau, t_1, t_2)$ be the number of calls served after a waiting time less than or equal to τ during

interval $[t_1,t_2]$. Let $N(t_1,t_2)$ be the total number of calls counted during interval $[t_1,t_2]$ and $L(\tau,t_1,t_2)$ be the number of calls having abandoned after a waiting time smaller than or equal to τ during the same time interval. Since the arrival and service times of calls are not known but are random, the service level in a given time period $[t_1,t_2]$ will be a random variable and a formula of service level in the time interval $[t_1,t_2]$ can be

$$f_S^1(\tau,t_1,t_2) = \frac{S(\tau,t_1,t_2)}{N(t_1,t_2) - L(\tau,t_1,t_2)} \,. \tag{2.2}$$

This definition of service level (2.2) is used in our formulation with chance constraints. For any given fixed staffing of agents, no reliable formula or quick algorithm is available to estimate the distribution of service level; it can be estimated based only by long (stochastic) simulation. An example of chance-constraint on the service level is, for example, the probability that at least 95% of calls are answered within $\tau = 2$ seconds in a given time period is equal to or greater than 85%. This constraint is used for the model of the emergency call center 911 in Montreal in our numerical experiment.

Another formula of service level defines the long-term fraction of calls whose time in queue is no larger than a given threshold, that is:

$$f_S^2(\tau,t_1,t_2) = \frac{\mathbb{E}[S(\tau,t_1,t_2)]}{\mathbb{E}[N(t_1,t_2) - L(\tau,t_1,t_2)]} \,, \tag{2.3}$$

where $\mathbb{E}$ denotes the mathematical expectation.

In this definition, the numerator is the average of calls answered within τ and the denominator is the average number of arrival calls (without abandonments), over an infinite time horizon. This service level define in (2.3) is equal to the fraction of calls answered within τ over an infinite number of independent and identically distributed (i.i.d.) copies of intervals $[t_1,t_2]$. It was used in most previous articles on staffing and scheduling optimization (e.g., Atlason et al. [1], Avramidis et al. [3], Avramidis et al. [4], etc). Multiple measures of SL are of interest: for a given time period of a day, for a given call type, for a given combination of call type and period, aggregated over the whole day and all call types, and so on. A typical constraint on the SL is, for example, that 80% of calls are answered within $\tau = 20$ seconds. In these contexts, they approximate f_S^2 by

simulations, the expectations being estimated by the sample averages.

Here are two alternative definitions of SL. Note that for each way to compute SL, we also can distinguish two situations: the expected value over an infinite time horizon or the random variable in a given time period. We here focus on the service level calculated in a given time period; the extension to infinite horizon is direct. The first alternative considers abandoned calls who waited less than or equal to τ as "good served calls":

$$f_S^3(\tau, t_1, t_2) = \frac{S(\tau, t_1, t_2) + L(\tau, t_1, t_2)}{N(t_1, t_2)} \,. \tag{2.4}$$

Another alternative formula considers all abandonments as "badly served calls":

$$f_S^4(\tau, t_1, t_2) = \frac{S(\tau, t_1, t_2)}{N(t_1, t_2)} \,. \tag{2.5}$$

Another performance measure is the *average waiting time*. It is the average (or mean) length of time a customer waited to have a service. The average waiting time is calculated by diving the total number of waiting time of all calls, by the total number of calls during the time period. Similar to the service level, we also have many definitions of average waiting time. We give two formulas for this measure, the former being computed over a given time period and the latter being defined for a long run.

A formula of average waiting time over a given time period $[t_1, t_2]$ is:

$$f_W^1(t_1, t_2) = \frac{W(t_1, t_2)}{N(t_1, t_2)} \,, \tag{2.6}$$

where $W(t_1, t_2)$ is the sum of waiting times of calls (served or abandoned) counted during time interval $[t_1, t_2]$. The average waiting time in this definition is a random variable, and is used in our formulations with chance constraints. An example of the chance constraint with average waiting time is that the probability that the average waiting time in a given time period does not exceed 2 seconds is no smaller than 85%.

An alternative definition represents the average waiting time within a given time $[t_1, t_2]$ in the long run:

$$f_W^2(t_1, t_2) = \frac{\mathbb{E}[W(t_1, t_2)]}{\mathbb{E}[N(t_1, t_2)]} \,. \tag{2.7}$$

The long term expected waiting time f_W^2 can be estimated by simulations, by dividing the average sum of waiting times by the average number of arrivals.

For other performance measures discussed in the following, we also distinguish them in two definitions, the expected value over the long run and the random variable in a given time interval. We only describe here their definitions as random variables.

An important measure is the *abandonment ratio*, defined as:

$$f_A(t_1,t_2) = \frac{A(t_1,t_2)}{N(t_1,t_2)}\ , \tag{2.8}$$

where $A(t_1,t_2)$ is the total number of abandonments. Abandonments usually mean the loss of potential customers.

The effectiveness of a call center is also often measured by the occupation rates of agents. Let y_i be the number of agents in the group i, T be the time horizon covered by the measure and $G_i(t) \leq y_i$ be the number of agents of group i occupied in the time $0 \leq t \leq T$. The *occupancy ratio* is defined by the proportion of agents occupied during the period of length T:

$$f_{O,i}(T) = \frac{1}{y_i T}\int_0^T G_i(t)\mathrm{d}t\ . \tag{2.9}$$

2.3 Description of the model

We consider a *telephone call center* where different types of calls arrive at random and different groups of agents answer these calls. The calls arrive according to arbitrary stochastic processes that could be non-stationary, and perhaps doubly stochastic (see, e.g., Avramidis et al. [2]). Arriving calls that find all servers occupied line up in an infinite buffer queue. Arrivals are served in a FCFS order.

The goal is to minimize the operating cost of the center under a set of constraints on the QoS. The day is divided into periods (e.g., 30 minutes or one hour each). The objective function is the sum of costs of all agents, where the *cost* of an agent is a deterministic function of its set of skills.

In this context, we are interested in chance constraints on the service levels and daily average waiting times. Such constraints can be imposed per call type, per period, and

globally, with different thresholds. The convergence of our sample problem and our algorithms can be applied similarly when adding more constraints on other types of performance measures, e.g., abandonment ratio, occupancy ratio, etc.

Our model of a call center is composed of a set of K call types, labelled from 1 to K, and I agent types, labelled from 1 to I. Agent type i has the skill set $S_i \subseteq \{1,...,K\}$. The day is divided into P periods of given length, labelled from 1 to P. The staffing vector is $y = (y_{1,1},...,y_{1,P},...,y_{I,1},...,y_{I,P})$ where $y_{i,p}$ is the number of agents of type i available in period p. Given y, let $S_{k,p}(\tau_{k,p},y)$ be the fraction of calls of type k answered within $\tau_{k,p}$ seconds during period p (the *service level*); let $S_p(\tau_p,y)$ be the fraction of calls answered within τ_p seconds during period p, let $S_k(\tau_k,y)$ be the fraction of calls of type k answered within τ_k seconds during the day; let $S_0(\tau_0,y)$ be the fraction of all calls answered within τ_0 seconds during the day; let $W_{k,p}(y)$ be the average waiting time for calls of type k during period p; and let $W_p(y)$ be the average waiting time for all calls arriving in period p; let $W_k(y)$ be the average waiting time of calls of type k during the day; let $W_0(y)$ be the average waiting time of all calls during the day. All of these are random variables, whose distributions depend on the entire staffing. Suppose that the constraints are of the forms: *the probabilities that service level and average waiting time are satisfied are no smaller than some given thresholds*.

The service level constraints are of the form:

$$\begin{aligned}
\mathbb{P}[S_{k,p}(\tau_{k,p},y) \geq s_{k,p}] &\geq r_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P\ ; \\
\mathbb{P}[S_p(\tau_p,y) \geq s_p] &\geq r_p \qquad \text{for } 1 \leq p \leq P\ ; \\
\mathbb{P}[S_k(\tau_k,y) \geq s_k] &\geq r_k \qquad \text{for } 1 \leq k \leq K\ ; \\
\mathbb{P}[S_0(\tau_0,y) \geq s_0] &\geq r_0\ ;
\end{aligned}$$

where $s_{k,p}$, s_p, s_k, s_0 are targets of service level and $r_{k,p}$, r_p, r_k, r_0 are given constants in $(0,1)$.

The average waiting time constraints are of the form:

$$\begin{aligned}
\mathbb{P}[W_{k,p}(y) \leq w_{k,p}] &\geq v_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P\ ; \\
\mathbb{P}[W_p(y) \leq w_p] &\geq v_p \qquad \text{for } 1 \leq p \leq P\ ; \\
\mathbb{P}[W_k(y) \leq w_k] &\geq v_k \qquad \text{for } 1 \leq k \leq K\ ; \\
\mathbb{P}[W_0(y) \leq w_0] &\geq v_0\ ;
\end{aligned}$$

where $w_{k,p}$, w_p, w_k, w_0 are targets of average waiting time and $v_{k,p}$, v_p, v_k, v_0 are given constants in $(0,1)$.

A *shift* is a time pattern that specifies the periods in which an agent is available to handle calls. In practice, it is defined by its *start period* (the period in which the agent starts working), *break periods* (the periods when the agent stops working, for instance, morning and afternoon coffee breaks, as well as a longer lunch break), and *end period* (the period when the agent finishes the workday).

Let $\{1,...,Q\}$ be the set of all admissible shifts. To simplify the exposition, we assume that this set is the same for all agent types; this assumption could easily be relaxed if needed, by introducing specific shift sets for each agent type. The admissible shifts are specified via a $P \times Q$ matrix B_0 whose element (p,q) is $B_{p,q} = 1$ if an agent with shift q works in period p, and 0 otherwise. A vector $x = (x_{1,1},...,x_{1,Q},...,x_{I,1},...,x_{I,Q})$, where $x_{i,q}$ is the number of agents of type i working shift q, is a *schedule*. The cost vector is $c = (c_{1,1},...,c_{1,Q},...,c_{I,1},...,c_{I,Q})$, where $c_{i,q}$ is the cost of an agent of type i with shift q. To any given shift vector x, there corresponds the staffing vector $y = Bx$, where B is a block-diagonal matrix with I identical blocks B_0, if we assume that each agent of type i works as a type-i agent for the entire shift. We make the following natural assumption that every period is covered by at least one shift.

Assumption 2.3.1. *For every period p there is at least one shift q such that $B_{p,q} = 1$.*

Calculating the cost function is usually relatively straightforward. We can calculate the cost of each group in each shift, and multiply by the number of agents of corresponding group working in the shift to get the overall cost. The cost function is defined by:

$$f(y) = \min c^t x = \sum_{i=1}^{I} \sum_{q=1}^{Q} c_{i,q} x_{i,q}$$

subject to:

$$\begin{aligned} Bx &\geq y; \\ x &\geq 0 \text{ and integer .} \end{aligned} \tag{2.10}$$

It follows by Assumption 2.3.1 that (2.10) is feasible for any y. The value $f(y)$ gives the minimum cost set of shifts that can cover the desired work requirements vector y. We make the following assumption on the cost vector.

Assumption 2.3.2. *The cost vector c is positive.*

According to Assumption 2.3.2, since c is positive and moreover, the entries in B are either 0 or 1, the α-level set of f,

$$\{y \in \mathbb{N}\text{: } \exists x \geq 0 \text{ and integer, } Bx \geq y,\ c^T x \leq \alpha\},$$

is finite, for any $\alpha \in \mathbb{R}$.

Define:

$h^1_{k,p}(y) = \mathbb{P}[S_{k,p}(\tau_{k,p}, y) \geq s_{k,p}] - r_{k,p}$ for $k = 1,...,K$, $p = 1,...,P$;

$h^1_{0,p}(y) = \mathbb{P}[S_p(\tau_p, y) \geq s_p] - r_p$ for $p = 1,...,P$;

$h^1_{k,0}(y) = \mathbb{P}[S_k(\tau_k, y) \geq s_k] - r_k$ for $k = 1,...,K$;

$h^1_{0,0}(y) = \mathbb{P}[S_0(\tau_0, y) \geq s_0] - r_0$;

and

$h^2_{k,p}(y) = \mathbb{P}[W_{k,p}(y) \leq w_{k,p}] - v_{k,p}$ for $k = 1,...,K$, $p = 1,...,P$;

$h^2_{0,p}(y) = \mathbb{P}[W_p(y) \leq w_p] - v_p$ for $p = 1,...,P$;

$h^2_{k,0}(y) = \mathbb{P}[W_k(y) \leq w_k] - v_k$ for $k = 1,...,K$;

$h^2_{0,0}(y) = \mathbb{P}[W_0(y) \leq w_0] - v_0$;

Let $g : \mathbb{R}^{IP} \to \mathbb{R}^{2(KP+K+P+1)}$ be a function defined by:

$$g = (h^1_{0,0}, h^1_{0,1}, ..., h^1_{0,P}, h^1_{1,0}, ..., h^1_{1,P}, ..., h^1_{K,0}, ..., h^1_{K,P},$$

$$h^2_{0,0}, h^2_{0,1}, ..., h^2_{0,P}, h^2_{1,0}, ..., h^2_{1,P}, ..., h^2_{K,0}, ..., h^2_{K,P})$$
$$\stackrel{\text{def}}{:=} (g_1, g_2, ..., g_{2(KP+K+P+1)}).$$

We are now ready to formulate the problem of minimizing scheduling costs subject to satisfying chance constraints in service level and average waiting time:

$$\min f(y)$$

subject to:

$$\begin{aligned} g(y) &\geq 0 \ ; \\ y &\geq 0 \text{ and integer.} \end{aligned} \tag{P0}$$

Note that problem (P0) is equivalent to the *scheduling problem*:

$$\min c^t x = \sum_{i=1}^{I} \sum_{q=1}^{Q} c_{i,q} x_{i,q}$$

subject to:

$$\begin{aligned} Bx &\geq y; \\ \mathbb{P}[S_{k,p}(\tau_{k,p}, y) \geq s_{k,p}] &\geq r_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P \ ; \\ \mathbb{P}[S_p(\tau_p, y) \geq s_p] &\geq r_p \qquad \text{for } 1 \leq p \leq P \ ; \\ \mathbb{P}[S_k(\tau_k, y) \geq s_k] &\geq r_k \qquad \text{for } 1 \leq k \leq K \ ; \\ \mathbb{P}[S_0(\tau_0, y) \geq s_0] &\geq r_0 \ ; \\ \mathbb{P}[W_{k,p}(y) \leq w_{k,p}] &\geq v_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P \ ; \\ \mathbb{P}[W_p(y) \leq w_p] &\geq v_p \qquad \text{for } 1 \leq p \leq P \ ; \\ \mathbb{P}[W_k(y) \leq w_k] &\geq v_k \qquad \text{for } 1 \leq k \leq K \ ; \\ \mathbb{P}[W_0(y) \leq w_0] &\geq v_0 \ ; \\ x, y &\geq 0 \text{ and integer .} \end{aligned} \tag{P1}$$

The *staffing problem* is a *relaxation* of the scheduling problem where we forget about the admissibility of schedules and just assume that any staffing is admissible. The

staffing cost vector in this setting is $d = (d_{1,1},...,d_{1,P},...,d_{I,1},...,d_{I,P})^t$ where $d_{i,p}$ is the cost of an agent of group i in period p. We have:

$$\min d^t y = \sum_{i=1}^{I} \sum_{p=1}^{P} d_{i,p} y_{i,p}$$

subject to:

$$\begin{aligned}
\mathbb{P}[S_{k,p}(\tau_{k,p}, y) \geq s_{k,p}] &\geq r_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[S_p(\tau_p, y) \geq s_p] &\geq r_p \qquad \text{for } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[S_k(\tau_k, y) \geq s_k] &\geq r_k \qquad \text{for } 1 \leq k \leq K \text{ ;} \\
\mathbb{P}[S_0(\tau_0, y) \geq s_0] &\geq r_0 \text{ ;} \\
\mathbb{P}[W_{k,p}(y) \leq w_{k,p}] &\geq v_{k,p} \qquad \text{for } 1 \leq k \leq K \text{ and } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[W_p(y) \leq w_p] &\geq v_p \qquad \text{for } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[W_k(y) \leq w_k] &\geq v_k \qquad \text{for } 1 \leq k \leq K \text{ ;} \\
\mathbb{P}[W_0(y) \leq w_0] &\geq v_0 \text{ ;} \\
y &\geq 0 \text{ and integer .}
\end{aligned} \tag{P2}$$

In the special case where we consider *one period at a time*, we have $d = (d_1,...,d_I)^t$ where d_i is the cost of an agent of type i, suppose that d is positive and $y = (y_1,...,y_I)^t$ where y_i is the number of agents of type i. In this context, we often assume that the system is in steady-state over the given period (but we may also assume arbitrary initial conditions). The optimization problem then reduces to:

$$\min d^t y = \sum_{i=1}^{I} d_i y_i$$

subject to:

$$\begin{aligned}
\mathbb{P}[S_k(\tau_k, y) \geq s_k] &\geq r_k \qquad \text{for } 1 \leq k \leq K \text{ ;} \\
\mathbb{P}[S_0(\tau_0, y) \geq s_0] &\geq r_0 \text{ ;} \\
\mathbb{P}[W_k(y) \leq w_k] &\geq v_k \qquad \text{for } 1 \leq k \leq K \text{ ;} \\
\mathbb{P}[W_0(y) \leq w_0] &\geq v_0 \text{ ;} \\
y &\geq 0 \text{ and integer .}
\end{aligned} \tag{P3}$$

In the case where we consider one call type, one agent group and P periods, we have $d = (d_1, ..., d_P)^t$ where d_p is the cost of an agent in period p and $y = (y_1, ..., y_P)^t$ where y_p is the number of agents in period p. The optimization problem then reduces to:

$$\min d^t y = \sum_{p=1}^{P} d_p y_p$$

subject to:

$$\begin{aligned}
\mathbb{P}[S_p(\tau_p, y) \geq s_p] &\geq r_p \qquad \text{for } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[S_0(\tau_0, y) \geq s_0] &\geq r_0 \text{ ;} \\
\mathbb{P}[W_p(y) \leq w_p] &\geq v_p \qquad \text{for } 1 \leq p \leq P \text{ ;} \\
\mathbb{P}[W_0(y) \leq w_0] &\geq v_0 \text{ ;} \\
y &\geq 0 \text{ and integer .}
\end{aligned} \tag{P4}$$

The underlying model is typically so complex that an algebraic expression for $g(y)$ can not be easily obtained. Therefore, simulation could be the only viable method for estimating $g(y)$. In the next section we formulate an optimization problem which is an approximate of (P0) obtained by replacing the probability values by sample averages and prove statements about the convergence of solutions of this sample problem to the solutions of the original problem (P0) as the sample size increases.

CHAPTER 3

SAMPLE AVERAGE APPROXIMATION OF THE CHANCE-CONSTRAINED STAFFING PROBLEM

The probabilities that the service level and average waiting time are satisfied in a given period involved in the constraints are estimated by simulation. Suppose we simulate the center N times, independently, over its P periods of operation. Let ω represent the source of randomness, i.e., the sequence of all independent $U(0,1)$ random variates that drive the successive simulation runs (regardless of their number). When simulating the call center for different values of a staffing level y, we assume that the same uniform random numbers are used for the same purpose for all values of y, for each day. This is implemented by using random number packages that provide multiple streams and substreams (L'Ecuyer [25], L'Ecuyer et al. [27]).

Suppose we perform N simulation runs to get the estimates of probabilities. We consider the distribution of the values of the service level and average waiting time over the individual runs. The *empirical service-level* of a simulation run, defined as the observed number of calls answered within the time limit divided by the total number of calls in this run, is a function of the staffing level y and of ω. We denote the empirical service level by the d-th replication by $\hat{S}^d_{N,k,p}(\tau_{k,p}, y, \omega)$ for call type k in period p; $\hat{S}^d_{N,p}(\tau_p, y, \omega)$ aggregated over period p; $\hat{S}^d_{N,k}(\tau_k, y, \omega)$ aggregated for all call type k; and $\hat{S}^d_{N,0}(\tau_0, y, \omega)$ aggregated overall. Similarly, the *empirical average waiting time* of a simulation run, defined as the total of waiting times of calls divided the total number of calls in this replication, is also a function of y and of ω. We denote the empirical average waiting time by the d-th replication by $\hat{W}^d_{N,k,p}(\tau_{k,p}, y, \omega)$ for call type k in period p; $\hat{W}^d_{N,p}(\tau_p, y, \omega)$ aggregated over period p; $\hat{W}^d_{N,k}(\tau_k, y, \omega)$ aggregated for all call type k; and $\hat{W}^d_{N,0}(\tau_0, y, \omega)$ aggregated overall. To compute these functions at different values of y, we simply use simulation with common random numbers, i.e., make sure that the same random numbers are used at the same place for all values of y (Law [24], Chapter 10). For simplicity of notations, we omit ω from the notations.

Let

$$\overline{h}^1_{k,p}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{S}^d_{N,k,p}(\tau_{k,p},y) \geq s_{k,p}]}{N} - r_{k,p} \qquad \text{for } k = 1,...,K,\ p = 1,...,P;$$
$$\overline{h}^1_{0,p}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{S}^d_{N,p}(\tau_p,y) \geq s_p]}{N} - r_p \qquad \text{for } p = 1,...,P;$$
$$\overline{h}^1_{k,0}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{S}^d_{N,k}(\tau_k,y) \geq s_k]}{N} - r_k \qquad \text{for } k = 1,...,K;$$
$$\overline{h}^1_{0,0}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{S}^d_{N,0}(\tau_0,y) \geq s_0]}{N} - r_0;$$

and

$$\overline{h}^2_{k,p}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{W}^d_{N,k,p}(y) \leq w_{k,p}]}{N} - v_{k,p} \qquad \text{for } k = 1,...,K,\ p = 1,...,P;$$
$$\overline{h}^2_{0,p}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{W}^d_{N,p}(y) \leq w_p]}{N} - v_p \qquad \text{for } p = 1,...,P;$$
$$\overline{h}^2_{k,0}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{W}^d_{N,k}(y) \leq w_k]}{N} - v_k \qquad \text{for } k = 1,...,K;$$
$$\overline{h}^2_{0,0}(y;N) = \frac{\sum_{d=1}^N \mathbb{I}[\hat{W}^d_{N,0}(y) \leq w_0]}{N} - v_0;$$

where $\mathbb{I}$ is the indicator function:

$$\mathbb{I}[A] = \begin{cases} 1 & \text{if the clause } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\overline{g} : \mathbb{R}^{IP} \times \mathbb{N} \to \mathbb{R}^{2(KP+K+P+1)}$ be the function of (y,N):

$$\overline{g} = (\overline{h}^1_{0,0}, \overline{h}^1_{0,1}, ..., \overline{h}^1_{0,P}, \overline{h}^1_{1,0}, ..., \overline{h}^1_{1,P}, ..., \overline{h}^1_{K,0}, ..., \overline{h}^1_{K,P},$$

$$\overline{h}^2_{0,0}, \overline{h}^2_{0,1}, ..., \overline{h}^2_{0,P}, \overline{h}^2_{1,0}, ..., \overline{h}^2_{1,P}, ..., \overline{h}^2_{K,0}, ..., \overline{h}^2_{K,P}).$$

$$\stackrel{\text{def}}{:=} (\overline{g}_1, \overline{g}_2, ..., \overline{g}_{2(KP+K+P+1)}).$$

We will replace the probabilities in the optimization problem by the averages in the sam-

ple problem, and we use these notations to formulate the Sample Average Approximation (SAA) problem:

$$\min f(y)$$

subject to:

$$\begin{aligned} \overline{g}(y;N) &\geq 0 \ ; \\ y &\geq 0 \text{ and integer.} \end{aligned} \tag{S1}$$

A similar formulation can be given for the other problems, replacing the probabilities on the left sides of the constrains by the sample means everywhere. The chance-constrained staffing problem for one call type, one skill agent, over a time interval divided into periods becomes:

$$\min d^t y = \sum_{p=1}^{P} d_p y_p$$

subject to:

$$\begin{aligned} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,p}(\tau_p, y) \geq s_p]}{N} &\geq r_p \qquad \text{for } 1 \leq p \leq P \ ; \\ \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,0}(\tau_0, y) \geq s_0]}{N} &\geq r_0 \ ; \\ \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,p}(y) \leq w_p]}{N} &\geq v_p \qquad \text{for } 1 \leq p \leq P \ ; \\ \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,0}(y) \leq w_0]}{N} &\geq v_0 \ ; \\ y &\geq 0 \text{ and integer} . \end{aligned} \tag{S2}$$

3.1 Almost Sure Convergence of Optimal Solutions of the Sample Average Approximation Problem

We insist on the fact that when ω is fixed, these sample problems are purely deterministic. These are the problems we solve, instead of the original problems (P0) to (S1).

We will study the convergence of the optimal solution of the sample problem to that of the original problem as $N \to \infty$, under the assumption that ω is really an infinite sequence of i.i.d. random variables. Some notations as well as the process of our proof

are similar with the proof in section 4 in Atlason et al. [1] with some variations. In that section, they prove the convergence of solutions of the sample average approximation to solutions of the original staffing problem. However, the constraints in their original problem are not chance constraints, they are in terms of infinite horizon service levels. Therefore, we have some modifications to adopt for our chance-constrained staffing problem. We are interested in the properties of the optimal solutions of (S1) as the sample size N gets large. By the suggestion in the proof of Atlason et al. [1], we use the Strong Law of Large Numbers (SLLN) to turn out that any optimal solution of (P0) that satisfies $g(y) > 0$, i.e., $g_j(y) > 0$ for all $1 \leq j \leq 2(KP+K+P+1)$, is an optimal solution of (S1) with probability 1 (w.p.1) as N goes to infinity.

We say that the property $A(N)$ holds for all N large enough w.p.1 if and only if $\mathbb{P}[\exists N_0 < \infty : \ A(N) \text{ holds } \forall N \geq N_0] = 1$.

The SLLN states that the sample average converges almost surely to the expected value:

$$\overline{X_n} = \frac{X_1 + ... + X_n}{n} \xrightarrow{\text{a.s}} \mu \text{ for } n \to \infty$$

where $X_1, X_2, ...$ is an infinite sequence of i.i.d. random variables with expected value $\mathbb{E}[X_1] = \mathbb{E}[X_2] = ... = \mu$ and $\mathbb{E}[|X_j|] < \infty$ for $j = 1, 2, ...$

That is, $\mathbb{P}[\lim_{n\to\infty} \overline{X_n} = \mu] = 1$.

We introduce some additional notations as in Atlason et al. [1]. Let

$$\overline{g}(y;\infty) := \lim_{N\to\infty} \overline{g}(y;N),$$
$$F^* := \text{the optimal value of (P0)}$$

and define the sets

$$Y^* := \text{the set of optimal solutions to (P0)},$$
$$Y_0^* := \{y \in Y^* : g(y) > 0\},$$
$$Y_1 := \{y \in \mathbb{Z}_+^{IP} : f(y) \leq F^*, g(y) \ngeq 0\},$$
$$Y_N^* := \text{the set of optimal solution of (S1)}.$$

Note that Y_1 is the set of solutions to (P0) that have the same or lower cost than an optimal solution, and satisfy all constraints except the probability constraints. We are concerned with solutions in this set since they could be feasible (optimal) for the sample problem (S1) if the difference between the sample average $\overline{g}(.;N)$ and g is sufficiently large. We show that when Y_0^* is not empty, $Y_0^* \subseteq Y_N^* \subseteq Y^*$ for all N large enough, w.p.1.

The sets Y^*, Y_N^* and Y_0^* are finite by Assumption 2.3.2. (The sets Y^*, Y_N^* and Y_0^* can be empty). Furthermore, if Y^* is nonempty then $F^* < \infty$ and then, again by Assumption 2.3.2, the set Y_1 is finite.

We start with two lemmas. The first one establishes properties of $\overline{g}(y;\infty)$ by repeatedly applying the SLLN. The second shows that solutions to (P0) satisfying $g(y) > 0$, and infeasible solutions, will be feasible and infeasible, respectively, w.p.1 for problem (S1) when N gets large. The only condition $g(y)$ has to satisfy is that it has to be finite for all $y \in \mathbb{Z}_+^{IP}$. That assumption is obviously satisfied because the values of probability functions never exceed 1.

Define

$$\|g\| = \max_{y \in \mathbb{Z}_+^{IP}} \|g(y)\|_\infty = \max_{y \in \mathbb{Z}_+^{IP}} \max_{j=1,\ldots,2(KP+K+P+1)} |g_j(y)|.$$

Lemma 3.1.1. *1. Suppose that $\|g(y)\|_\infty < \infty$ for some fixed $y \in \mathbb{Z}_+^{IP}$. Then $\overline{g}(y;\infty) = g(y)$ w.p.1.*

2. Suppose that $\|g\|_\infty < \infty$ and $\Gamma \subset \mathbb{Z}_+^{IP}$ is finite. Then $\overline{g}(y;\infty) = g(y)\ \forall y \in \Gamma$ w.p.1.

Proof. 1. For all $1 \le k \le K$, $1 \le p \le P$ and $1 \le d \le N$, we notice that:

$$\mathbb{E}[\mathbb{I}[\hat{S}^d_{N,k,p}(\tau_{k,p},y) \ge s_{k,p}]] = \mathbb{P}[\hat{S}^d_{N,k,p}(\tau_{k,p},y) \ge s_{k,p}] = \mathbb{P}[S_{k,p}(\tau_{k,p},y) \ge s_{k,p}],$$

$$\mathbb{E}[\mathbb{I}[\hat{S}^d_{N,p}(\tau_p,y) \ge s_p]] = \mathbb{P}[\hat{S}^d_{N,p}(\tau_p,y) \ge s_p] = \mathbb{P}[S_p(\tau_p,y) \ge s_p],$$

$$\mathbb{E}[\mathbb{I}[\hat{S}^d_{N,k}(\tau_k,y) \ge s_k]] = \mathbb{P}[\hat{S}^d_{N,k}(\tau_k,y) \ge s_k] = \mathbb{P}[S_k(\tau_k,y) \ge s_k],$$

$$\mathbb{E}[\mathbb{I}[\hat{S}^d_{N,0}(\tau_0,y) \ge s_0]] = \mathbb{P}[\hat{S}^d_{N,0}(\tau_0,y) \ge s_0] = \mathbb{P}[S_0(\tau_0,y) \ge s_0],$$

$$\mathbb{E}[\mathbb{I}[\hat{W}^d_{N,k,p}(y) \le w_{k,p}]] = \mathbb{P}[\hat{W}^d_{N,k,p}(y) \le w_{k,p}] = \mathbb{P}[W_{k,p}(y) \le w_{k,p}],$$

$$\mathbb{E}[\mathbb{I}[\hat{W}^d_{N,p}(y) \le w_p]] = \mathbb{P}[\hat{W}^d_{N,p}(y) \le w_p] = \mathbb{P}[W_p(y) \le w_p],$$

$$\mathbb{E}[\mathbb{I}[\hat{W}^d_{N,k}(y) \le w_k]] = \mathbb{P}[\hat{W}^d_{N,k}(y) \le w_k] = \mathbb{P}[W_k(y) \le w_k]$$

$$\mathbb{E}[\mathbb{I}[\hat{W}^d_{N,0}(y) \le w_0]] = \mathbb{P}[\hat{W}^d_{N,0}(y) \le w_0] = \mathbb{P}[W_0(y) \le w_0].$$

By using the SLLN:

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,k,p}(\tau_{k,p}, y) \ge s_{k,p}]}{N}] = \mathbb{P}[S_{k,p}(\tau_{k,p}, y) \ge s_{k,p}] \text{ w.p.1.}$$

Similarly, we have:

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,p}(\tau_p, y) \ge s_p]}{N}] = \mathbb{P}[S_p(\tau_p, y) \ge s_p] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,k}(\tau_k, y) \ge s_k]}{N}] = \mathbb{P}[S_k(\tau_k, y) \ge s_k] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,0}(\tau_0, y) \ge s_0]}{N}] = \mathbb{P}[S_0(\tau_0, y) \ge s_0] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,k,p}(y) \le w_{k,p}]}{N} = \mathbb{P}[W_{k,p}(y) \le w_{k,p}] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,p}(y) \le w_p]}{N} = \mathbb{P}[W_p(y) \le w_p] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,k}(y) \le w_k]}{N} = \mathbb{P}[W_k(y) \le w_k] \text{ w.p.1,}$$

$$\lim_{N\to\infty} \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{W}^d_{N,0}(y) \le w_0]}{N} = \mathbb{P}[W_0(y) \le w_0] \text{ w.p.1.}$$

So $\overline{g}_j(y;\infty) = g_j(y)$ w.p.1 for $j \in \{1,...,2(KP+K+P+1)\}$.

Then

$$\mathbb{P}[\overline{g}(y;\infty) = g(y)] \ge 1 - \sum_{j=1}^{2(KP+K+P+1)} \mathbb{P}[\overline{g_j}(y;\infty) \ne g_j(y)] = 1$$

by Boole's inequality (3.1), stated as follow:

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{N} \mathbb{P}(A_i). \tag{3.1}$$

2. Note that

$$\mathbb{P}[\overline{g}(y;\infty) = g(y)\colon \forall y \in \Gamma] \geq 1 - \sum_{y\in\Gamma} \mathbb{P}[\overline{g}(y;\infty) \neq g(y)] = 1$$

since Γ is finite. □

Lemma 3.1.2. *Suppose that $\|g\| < \infty$ and that Assumption 2.3.2 holds. Then*

1. $\overline{g}(y;N) \geq 0\, \forall y \in Y_0^*$ for all N large enough w.p.1.

2. All $y \in Y_1$ are infeasible for the SAA problem (S1) *for all N large enough w.p.1 if $F^* < \infty$.*

Lemma (3.1.2) proof is very similar to that of Lemma 3 in Atlason et al. [1].

Proof. 1. The result is trivial if Y_0^* is empty, so suppose it is not. Let

$$\delta = \min_{y\in Y_0^*} \min_{j=1,\ldots,2(KP+K+P+1)} g_j(y).$$

Then $\delta > 0$ by the definition of Y_0^*. Let

$$N_0 = \inf\{n_0 : \max_{y\in Y_0^*} \|\overline{g}(y;N) - g(y)\|_\infty < \delta,\ \forall N \geq n_0\},$$

with the infimum defined as $+\infty$ if the set is empty. Then $\overline{g}(y;N) \geq 0, \forall y \in Y_0^*, \forall N \geq N_0$. The set Y_0^* is finite, so $\lim_{N\to\infty} \overline{g}(y;n) = g(y), \forall y \in Y_0^*$ w.p.1 by part 2 of Lemma 3.1.1. Therefore $N_0 < \infty$ w.p.1.

2. The result is trivial if Y_1 is empty, so suppose it is not. Let

$$\delta = \min_{y\in Y_1} \max_{j=1,\ldots,2(KP+K+P+1)} \{-g_j(y)\}.$$

Then $\delta > 0$, since $g_j(y) < 0$, for at least one $j \in \{1,...,2(KP+K+P+1)\}, \forall y \in Y_1$. Let

$$N_1 = \inf\{n_1 : \max_{y \in Y_1} \|g(y) - \overline{g}(y;N)\|_\infty < \delta, \forall N \geq n_1\}$$

and then all $y \in Y_1$ are infeasible for (S1) for all $N \geq N_1$. The set Y_1 is finite by Assumption 2.3.2 and since $F^* < \infty$, so

$$\lim_{N \to \infty} \overline{g}(y;N) = g(y), \forall y \in Y_1$$

w.p.1 by part 2 of Lemma 3.1.1. Therefore, $N_1 < \infty$ w.p.1. □

Lemma 3.1.2 shows that all the "interior" optimal solutions for the original problem are feasible for the SAA problem w.p.1 as the sample size is large enough. Furthermore, all solutions that satisfy the common constraints of both problems, except the probability constraints, and have at most the same cost as an optimal solution, become infeasible for the SAA problem w.p.1. Therefore, Atlason et al. [1] prove an important result that *for a large enough sample size, an optimal solution for the SAA problem is indeed optimal for the original problem.*

Theorem 3.1.3. *Suppose that $\|g\| < \infty$ and that Assumption 2.3.2 holds. Then $Y_0^* \subseteq Y_N^*$ for all N large enough w.p.1. Furthermore, if Y_0^* is nonempty then $Y_0^* \subseteq Y_N^* \subseteq Y^*$ for all N large enough w.p.1.*

Applying this theorem in the specific case $Y_0^* = Y^* = \{y^*\}$, Atlason et al. [1] obtain the following corollary:

Corollary 3.1.4. *Suppose that $\|g\| < \infty$ and that Assumption 2.3.2 holds and that* (P0) *has a unique optimal solution, y^*, such that $g(y^*) > 0$. Then y^* is the unique optimal solution for* (P2) *for all N large enough w.p.1.*

3.2 Exponential Rate of Convergence of Optimal Solutions of the Sample Problems

In the previous subsection, we showed that we can expect to get an optimal solution for the original problem (P0) by solving the SAA problem (S1) if we choose a large sample size. In this section, we show that the probability of getting an optimal solution this way approaches one exponentially fast as we increase the sample size. We use large deviations theory and a result due to Dai et al. [15] to prove our statement. The following theorem is an intermediate result from Theorem 3.1 in Dai et al. [15]:

Theorem 3.2.1. *Let $H : \mathbb{R}^{IP} \times \Omega \to \mathbb{R}$ and assume that there exists $\gamma > 0, \theta_0 > 0$ and $\eta : \Omega \to \mathbb{R}$ such that*

$$|H(y,\omega)| \leq \gamma\eta(\omega),\ \mathbb{E}[e^{\theta\eta(\omega)}] < \infty,$$

for all $y \in \mathbb{R}^p$ and for all $0 \leq \theta \leq \theta_0$, where ω is a random element taking values in the space Ω. Then for any $\delta > 0$, there exists $a > 0$, $b > 0$ such that for all any $y \in \mathbb{R}^{IP}$

$$\mathbb{P}[|h(y) - \overline{h}(y,N)| \geq \delta] \leq ae^{-bN}$$

for all $N > 0$, where $h(y) = \mathbb{E}[H(y,\omega)]$ and $\overline{h}(y,N)$ is a sample mean of N independent and identically distributed realizations of $H(y,\omega)$.

Set

$$\begin{aligned}
H^1_{k,p}(y,\omega) &= \mathbb{I}[S_{N,k,p}(\tau_{k,p},y) \geq s_{k,p}] - r_{k,p} \qquad \text{for } 1 \leq k \leq K, 1 \leq p \leq P;\\
H^1_{0,p}(y,\omega) &= \mathbb{I}[S_{N,p}(\tau_p,y) \geq s_p] - r_p \qquad \text{for } 1 \leq p \leq P;\\
H^1_{k,0}(y,\omega) &= \mathbb{I}[S_{N,k}(\tau_k,y) \geq s_k] - r_k \qquad \text{for } 1 \leq k \leq K;\\
H^1_{0,0}(y,\omega) &= \mathbb{I}[S_{N,0}(\tau_0,y) \geq s_0] - r_0;\\
H^2_{k,p}(y,\omega) &= \mathbb{I}[W_{N,k,p}(y) \leq w_{k,p}] - v_{k,p} \qquad \text{for } 1 \leq k \leq K, 1 \leq p \leq P;\\
H^2_{0,p}(y,\omega) &= \mathbb{I}[W_{N,p}(y) \leq w_p] - v_p \qquad \text{for } 1 \leq p \leq P;\\
H^2_{k,0}(y,\omega) &= \mathbb{I}[W_{N,k}(y) \leq w_k] - v_k \qquad \text{for } 1 \leq k \leq K;\\
H^2_{0,0}(y,\omega) &= \mathbb{I}[W_{N,0}(y) \leq w_0] - v_0.
\end{aligned}$$

Since $|H^t_{k,p}(y,\omega)| \leq 2$ for all $t \in \{1,2\}$, $0 \leq k \leq K$, $0 \leq p \leq P$, for all $y \in \mathbb{R}^{IP}$, we can apply the previous theorem with $\gamma = 2$, any $\theta_0 > 0$ and $\eta = 1$. Before we prove the exponential rate of convergence, we restate Lemma 3.2.2 in Atlason et al. [1] which shows that for any N, $Y_0^* \subseteq Y_N^* \subseteq Y^*$ precisely when all the solutions in Y_0^* are feasible for the SAA problem and all infeasible solutions for (P0) that are equally good or better, i.e., are in the set Y_1, are also infeasible for (S1).

Lemma 3.2.2. *Let $N > 0$ be an arbitrary integer and let Y_0^* be nonempty. The properties*

1. $\overline{g}(y,N) \geq 0 \; \forall y \in Y_0^*$, *and*
2. $\overline{g}(y,N) < 0 \; \forall y \in Y_1$

hold if and only if $Y_0^* \subseteq Y_N^* \subseteq Y^*$.

The following theorem show that the probability of getting an optimal solution by solving SAA problem approaches one exponentially fast as we increase the sample size. The proof is similar to that of Theorem 3.2.3 in Atlason et al. [1].

Theorem 3.2.3. *Suppose that Assumption 2.3.2 holds and that Y_0^* is nonempty. Then there exist $\alpha > 0$, $\beta > 0$ such that*

$$\mathbb{P}[Y_0^* \subseteq Y_N^* \subseteq Y^*] \geq 1 - \alpha e^{-\beta N}$$

Proof. Define

$$\begin{aligned}
\delta_1 &:= \min_{y \in Y_0^*} \min_{j \in \{1,\ldots,2(KP+K+P+1)\}} \{g_j(y)\}, \\
j(y) &:= \operatorname*{arg\,max}_{j \in \{1,\ldots,2(KP+K+P+1)\}} \{-g_j(y)\}, \\
\delta_2 &:= \min_{y \in Y_1} \{-g_{j(y)}(y)\}, \\
\delta &:= \min\{\delta_1, \delta_2\}
\end{aligned}$$

Here $\delta_1 > 0$ is the minimal slack value in the constraints $g(y) \geq 0$ for any solution $y \in Y_0^*$. Similarly, $\delta_2 > 0$ is the minimal violation in the constraints $g(y) \geq 0$ induced by

any solution $y \in Y_1$. Thus,

$$\begin{aligned}
\mathbb{P}[Y_0^* \subseteq Y_N^* \subseteq Y^*] &= \mathbb{P}[(\overline{g}(y,N) \geq 0 \,\forall y \in Y_0^*) \cap (\overline{g}(y,N) < 0 \,\forall y \in Y_1)] && (3.2)\\
&= 1 - \mathbb{P}[(\exists y \in Y_0^* \text{ s.t. } \overline{g}(y;N) < 0) \cup (\exists y \in Y_1 \text{ s.t. } \overline{g}(y;N) \geq 0)]\\
&\geq 1 - \sum_{y \in Y_0^*} \sum_{j=1}^{2(KP+K+P+1)} \mathbb{P}[\overline{g}_j(y;N) < 0] - \sum_{y \in Y_1} \mathbb{P}[\overline{g}(y;N) \geq 0] && (3.3)\\
&\geq 1 - \sum_{y \in Y_0^*} \sum_{j=1}^{2(KP+K+P+1)} \mathbb{P}[|\overline{g}_j(y;N) - g_j(y)| \geq \delta]\\
&\quad - \sum_{y \in Y_1} \mathbb{P}[|\overline{g}_{j(y)}(y;N) - g_{j(y)}(y)| \geq \delta] && (3.4)\\
&\geq 1 - \sum_{y \in Y_0^*} \sum_{j=1}^{2(KP+K+P+1)} a_j e^{-b_j N} - \sum_{y \in Y_1} a_{j(y)} e^{-b_{j(y)} N} && (3.5)\\
&\geq 1 - \alpha e^{-\beta N}.
\end{aligned}$$

Here

$$\alpha = \#Y_0^* \sum_{j=1}^{2(KP+K+P+1)} a_j + \sum_{y \in Y_1} a_{j(y)}$$

and

$$\beta = \min_{j=1,\ldots,2(KP+K+P+1)} b_j$$

where $\#Y_0^*$ is the cardinality of the set Y_0^*.

The sets Y_0^* and Y_1 are finite by Assumption 2.3.2, so $\alpha < \infty$. Equation (3.2) follows by Lemma 3.2.2. Equation (3.3) is Boole's inequality (3.1). Equation (3.4) follows since $\mathbb{P}[\overline{g}(y;N) \geq 0] \leq \mathbb{P}[\overline{g}_{j(y)}(y;N) \geq 0]$ and $g_j(y) \geq \delta_1 \geq \delta$ for $y \in Y_0^*$ and $g_{j(y)}(y) \geq \delta_2 \geq \delta$ for $y \in Y_1$. Finally, (3.5) follows from Theorem 3.2.1. □

Motivated by the results of this chapter, we would like to propose an algorithm to solve the sample chance-constrained scheduling problem (S1). However, in this thesis, we only concentrate on solving the staffing problem for the restricted case where *all agents are identical and can answer all call types*.

CHAPTER 4

SIMULATION METHODS

In this chapter, we propose three algorithms to solve our chance-constrained staffing problem where all agents are identical and can serve all call types, over a time interval divided into periods. This could correspond to one day divided into 48 half-hour periods, for example, in the situation where the call center opens 24 hours a day. In order to simplify, we suppose that all the value costs $d_p = 1$ for all $1 \leq p \leq P$. The general idea is to replace the problem (P4) by a *sample* version (S2), and then use simulation methods to solve this sample problem.

4.1 General idea for simulation algorithms

We propose three simulation-based optimization algorithms to solve the sample problem. All of them are rooted in the same idea, with some variations in the implementation. They can be decomposed in five stages, as described below:

Stage 1: `Initialize` We can choose an arbitrary initial staffing level. Two specific strategies are considered below. The simplest way is to start with a staffing equal to 0 for all periods. We can also choose an initial staffing level by using the Erlang C formula since in some cases, we may expect that Erlang C gives a staffing level which is close to a good solution. Note that Erlang C gives a staffing level satisfying the constraints on the expected service level in the long run.

With the initial staffing, there may exist some chance constraints in our sample problem (S2) that are satisfied while others are not. One natural approach would be to increase the staff number in some periods in which the constraints on the service level or average waiting time are violated, until these constraints are satisfied.

Stage 2: `Increase` We consider the periods in which the constraints are not satisfied, and increase the number of staffs in these periods until the constraints in these periods are satisfied.

Stage 3: `Decrease` After stage 2, all constraints in periods are satisfied. However, we can sometimes decrease the number of agents in several periods such that the constraints in these periods are still satisfied. In stage 3, we decrease the number of agents as much as possible, under the condition that the constraints in the individual periods are still satisfied.

Stage 4: `Increase-Last` We consider the constraints over the whole day. These constraints may be not satisfied. We continue increasing the staffing level until the constraints over the whole day are satisfied.

Stage 5: `Correction` As far as we know, changing the staffing in one period can change the performance (such as service level, etc.) in other periods as well. Atlason et al. [1] present an example showing that the staffing level in one period can have a considerable effect on the service level in another period. In that example, they explain the reason why the service level depends on the staffing level in the previous period. That is because a low staffing level in an earlier period results in a queue build-up, which increases waiting in the next period. They also explain the reason why the staffing level in a later period affects the service level in an earlier period, is that arrival calls in the earlier period may still be waiting at the beginning of the next period and thus are served earlier if there are more servers in that period. In some call centers, e.g., the emergency 911 call center, this effect is very small because there is rarely a queue in the system (the agents are not very busy), but in general, this effect could be very important. Since our algorithms are based on changing the number of staffing in periods, the manner and the order of periods of changing the number of agents may have noticeable affect on the results.

Therefore, to improve the quality of solutions, after the four stages, we add another stage, say stage `Correction`. In this stage, we consider all periods, from the first period. For each period, we try to decrease the number of agents in this period as much as possible, provided that the staffing level is still feasible for the sample problem.

In the next section, we will describe the three algorithms in more details as well as the differences between them.

4.2 Simulation algorithm 1: CCS1

The original method CCS1 is very simple to implement.

In stage 2, we consider the periods in which the constraints are not satisfied. For each of them, we increase the number of agents in this period by adding one unit in each iteration, until the constraints in this period are satisfied. After stage 2, all constraints in periods are satisfied.

In stage 3, in each iteration, we consider only one period, from the first period. We try to decrease the number of agents in this period as much as possible provided that all constraints in this period are still satisfied. Then, we consider the next period and repeat this process.

If the constraints in the whole day are not satisfied by using staffing level obtained from stage 3, in stage 4, we will increase the staffing levels until these constraints are satisfied. We may have plenty of choices to choose the periods in which we increase the number of agents. Here are several examples.

- At each iteration, we choose the period with the smallest service level, and add one agent in this period. After each time adding new agent and running simulation, we check if the constraints over the whole day are satisfied. This stage is stopped as soon as these constraints are satisfied.
- We consider the differences between the estimations of the probabilities that the constraints on the SL are satisfied and the target of the probabilities in all periods. The number of agents in the period with the lowest difference would be increased. For more detail, we consider constraints on the SL in all periods $p = 1, ..., P$. After setting any new staffing level and running simulations, we can compute:

$$\frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}_{N,p}^{d}(\tau_p, y) \geq s_p]}{N}.$$

We denote:

$$\Delta_p = \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,p}(\tau_p, y) \geq s_p]}{N} - r_p \text{ for all } p = 1, ..., P.$$

In stage 4, the number of agents in the period in which Δ_p is lowest will be increased. In the numerical experiment, we use this method to increase the number of agents. It is also expressed in details of the algorithm CCS1 below.

4.3 Simulation algorithm 2: CCS2

In CCS1, in stage 2, stage 3 and stage 4, we change the number of agents by at most one unit in a single period at each iteration. After each change of the number of agents, we perform a simulation. This approach can be time-consuming as it can require many simulations. In order to save computational time, in CCS2, we change (increase and decrease) the number of agents in several periods at the same time, i.e., in each iteration, we will increase or decrease the number of agents by at most one unit in each period, but *in all selected periods (where increase is required or decrease appears to be acceptable) simultaneously*.

We have observed a significant decrease of the required CPU time when implementing these modifications. We can see it in the next sections.

4.4 Simulation algorithm 3: CCS3

In the two above algorithms, the numbers of agents in each periods are changed by increasing or decreasing only one unit each time. Algorithm CCS3 is a modified version of CCS2, as we now use the bisection on the number of agents in stages 2 and 3. If the difference between the optimal solution and the initial solution is large, this approach can reduce the required CPU time, as the number of increases or decreases may be lower. The stage 4 is the same as for CCS1.

Algorithm 1 Algorithm 1

Require: P and other data for the model (arrival process, service time distributions,...).
Ensure: an estimation of an optimal solution y .
Begin
  Initialization
  $P \leftarrow$ Number of periods;
  $N \leftarrow$ Number of replications;
  $y = \{y_1, \ldots, y_p\}$; // *Initial solution*
  // Increase
  for $p : 1 \rightarrow P$ **do**
    while y_p does not satisfy the constraints in period p **do**
      $y_p \leftarrow y_p + 1$;
      Simulation with new staffings y;
    end while
  end for
  // Decrease
  for $p : 1 \rightarrow P$ **do**
    while y_p satisfies the constraints in period p **do**
      $y_p \leftarrow y_p - 1$;
      Simulation with new staffings y;
    end while
    $y_p \leftarrow y_p + 1$;
  end for
  // Increase-Last
  $\Delta = (\Delta_1, ..., \Delta_P)$;
  for $p : 1 \rightarrow P$ **do** $\Delta_p = \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,p}(\tau_p, y) \geq s_p]}{N} - r_p$;
  end for
  while y does not satisfy the constraints over the whole day **do**
    $k \leftarrow \arg\min_{p=1,\ldots,P} \Delta_p$;
    $y_k \leftarrow y_k + 1$;
    Simulation with new staffings y;
  end while
  // Correction
  for $p : 1 \rightarrow P$ **do**
    while y is feasible **do**
      $y_p \leftarrow y_p - 1$;
      Simulation with new staffings y;
    end while
    $y_p \leftarrow y_p + 1$;
  end for
End

Algorithm 2 Algorithm 2

Require: P and other data for the model (arrival process, service time distributions,...).
Ensure: an estimation of an optimal solution y .
Begin
    Initialization
    $P \leftarrow$ Number of periods;
    $N \leftarrow$ Number of replications;
    $y = \{y_1, \ldots, y_p\}$; *// Initial solution*
    //Increase
    $u = (u_1, ..., u_P) = (1, ..., 1)$;
    repeat
        for $p : 1 \rightarrow P$ **do**
            if y_p satisfies the constraints in period p **then**
                $u_j \leftarrow 0$;
            end if
            $y_p \leftarrow y_p + u_p$;
        end for
        Simulation with new staffings y;
    until $u = 0$;
    // Decrease
    $u = (u_1, ..., u_P) = (1, ..., 1)$;
    while $u \neq 0$ **do**
        for $p : 1 \rightarrow P$ **do**
            $y_p \leftarrow y_p - u_p$;
        end for
        Simulation with new staffings y;
        for $p : 1 \rightarrow P$ **do**
            if y_p does not satisfy the constraints in period p **then**
                $y_p \leftarrow y_p + u_p$; $u_p \leftarrow 0$;
            end if
        end for
    end while
    Simulation with new staffings y;
    // Increase-Last
    $u = (u_1, ..., u_P) = (1, ..., 1)$;
    while y do not satisfy the constraints over the whole day **do**
        for $p : 1 \rightarrow P$ **do**
            $y_p \leftarrow y_p + u_p$;
        end for
        Simulation with new staffings y;
    end while

▷ *Continue in next page ...*

```
42:     Sorting y by decreasing order of service level {y_{k_1}, ..., y_{k_P}};
43:     t = 1;
44:     while  y is feasible do
45:         y_{k_t} ← y_{k_t} − 1;
46:         Simulation with new staffings y;
47:         t ← t + 1;
48:     end while
49:     y_{k_{t−1}} ← y_{k_{t−1}} + 1;
50:     // Correction
51:     for p : 1 → P do
52:         while y is feasible do
53:             y_p ← y_p − 1;
54:             Simulation with new staffings y;
55:         end while
56:         y_p ← y_p + 1;
57:     end for
58: End
```

4.5 Analysis of the algorithms

Proposition 4.5.1. *Suppose that the sample problem* (S2) *is feasible. Then all three algorithms terminate at feasible solutions, in a finite number of iterations.*

Proof. Suppose that $y^* = (y_1^*, ..., y_P^*)$ is a feasible solution of the sample problem (S2), that $y_0 = (y_{01}, ..., y_{0P})$ is an initial staffing level, and that our algorithms do not stop after a finite number of iterations.

Assume also that the stage `Increase` do not stop after a finite number of iterations, i.e., our algorithms cannot find a staffing level which satisfies the constraints in all periods. However, for each $1 \leq p \leq P$, after a finite numbers of increases of the staffing in the period p, the number of agents in this period will be equal or greater to y_p^*. Therefore, the three algorithms can always find solutions which satisfy all the constraints for any period after a finite number of iterations.

In the stage `Decrease`, we decrease the staffing in all periods such that they still satisfy the constraints in all periods. Since the number of agents in each period is non-negative, this stage terminates after a finite number of iterations.

Suppose now that the stage `Increase-Last` do not stop after a finite number of

Algorithm 3 Algorithm 3

Require: P and other data for the model (arrival process, service time distributions,...).
Ensure: an estimation of an optimal solution y .
Begin
    Initialization
    $P \leftarrow$ Number of periods;
    $N \leftarrow$ Number of replications;
    $y = (y_1, \ldots, y_p)$; *// Initial solution*
    $x = (x_1, ..., x_P)$;
    // Increase
    $u = (u_1, ..., u_P) = (1, ..., 1)$;
    repeat
        for $p : 1 \rightarrow P$ **do**
            $y_p \leftarrow y_p + u_p$;
        end for
        for $p : 1 \rightarrow P$ **do**
            if y satisfies the constraints in period p **then**
                if $u_p \neq 0$ **then**
                    $x_p = y_p - u_p$;
                end if
                $u_p \leftarrow 0$;
            else
                $u_p \leftarrow 2 * u_p$;
            end if
        end for
        Simulation with new staffings y;
    until $u = 0$;
    // Decrease
    repeat
        for $j : 1 \rightarrow P$ **do**
            $\alpha_p \leftarrow \lfloor (x_p + y_p)/2 \rfloor$;
        end for
        Simulation with the staffings $(\alpha_1, ..., \alpha_P)$;
        for $p : 1 \rightarrow P$ **do**
            if α_p satisfies the constraints in period p **then**
                $y_p \leftarrow \alpha_p$;
            else
                $x_p \leftarrow \alpha_p$;
            end if
        end for
    until $(y_p - x_p \leq 1$ for all $p)$;

▷ *Continue in next page ...*

// Increase-Last
$\Delta = (\Delta_1, ..., \Delta_P)$;
for $p : 1 \to P$ **do** $\Delta_p = \frac{\sum_{d=1}^{N} \mathbb{I}[\hat{S}^d_{N,p}(\tau_p, y) \geq s_p]}{N} - r_p$;
end for
while y does not satisfy the constraints over the whole day **do**
    $k \leftarrow \arg\min_{j=1,...,P} \Delta_p$;
    $y_k \leftarrow y_k + 1$;
    Simulation with new staffings y;
end while
// Correction
for $p : 1 \to P$ **do**
    while y is feasible **do**
        $y_p \leftarrow y_p - 1$;
        Simulation with new staffings y;
    end while
    $y_p \leftarrow y_p + 1$;
end for
End

iterations, i.e., we can not increase the staffings to satisfy the constraints in the whole day. However, after a finite number of increases, we will obtain a staffing level $y = (y_1, ..., y_P)$ such that $y_p \geq y^*_p$ for all $1 \leq p \leq P$. Thus, this staffing level satisfies the constraints over the whole day. Therefore, the stage `Increase-Last` stops after a finite number of iterations.

Similarly, the stage `Correction` also terminates after a finite number of iterations.

□

In conclusion, all three algorithms terminate after finite numbers of iterations. Obviously, the three algorithms return staffing levels which satisfy all the constraints of our sample problem (S2), so they deliver upper bounds for the cost of our sample problem (S2). Moreover, in our algorithms, in the stage `Decrease`, we try to decrease the number of agents as much as possible, and the stage `Increase-Last` stops as soon as we find a staffing level which satisfies the constraints over the whole day. After that, in the stage `Correction`, we try to reduce the number of agents in all periods as much as possible, provided that we still obtain feasible solutions. Therefore, we could expect that

our algorithms give good heuristic solutions for our sample problem.

As discussed in Chapter 1, in the staffing problems where the service level constraints are on long term averages, there are many methods to solve these problems. We are also interested in applying these methods for our chance-constrained staffing problem. One of them is the *cutting plane method*, that we discuss in the next section. Compared with this method, the three methods we proposed above are much easier to implement and do not require a linear program solver.

Among the three algorithms, the algorithm CCS1 is simplest, but the computational time of running this algorithm is very large, compared with two others. Algorithm CCS2 is the best to save the running time in case the difference between the initial staffing and the optimal solution is small. Otherwise, the algorithm CCS3 is the best choice.

4.6 Cutting plane method for chance-constrained staffing problem

In Chapter 1, we have mentioned various algorithms to solve the staffing problems subject to the constraints in terms of infinite horizon service levels. One of them is the cutting plane method (Atlason et al. [1], Ceẓik and L'Ecuyer [12], etc.). This cutting plane method applies to minimization problems where both the objective function and feasible region (of the continuous relaxation of the integer problem) need to be convex. Atlason et al. [1] use this method to optimize the schedule of agents in a single-call-type and single-skill call center. Ceẓik and L'Ecuyer [12] extend this method to the multiskill setting. We think about applying this cutting plane method for our chance-constrained problem in a single skill call center. The cost in our problem is linear and we will assume that the probability that the constraint on the service level, or average waiting time, is satisfied, for each period, is a concave function. We relax the nonlinear probability constraints of (S2) to convert the chance-constrained staffing problem into a linear integer problem. We then solve the linear integer problem and run a simulation with the staffing levels obtained from the solution. If the probabilities meet the constraints as approximated by the sample average then we stop with an optimal solution to (S2). If a probability constraint is violated then we add a linear constraint to the relaxed problem that

eliminates the current solution but does not exclude any feasible solution to the sample problem. The procedure is then repeated. Similar to the cutting plane method to solve the staffing problem in Atlason et al. [1], the cutting plane method for chance-constrained staffing problems uses simulations to generate the cuts and evaluate the function values instead of having an algebraic form for the function and using analytically determined gradients to generate the cuts. We would expect that this is an interesting direction for our future research. Hopefully, this cutting plane method would solve the staffing problem with a single call type very quickly and may return good results, and we can extend this method for the schedule problem for call centers with chance constraints. However, this method requires a linear program solver, so it would be more difficult to implement than our three methods discussed above.

4.7 Out-of-sample analysis

After obtaining a final staffing level for the sample problem, this solution should be assessed for accuracy using an independent (out-of-sample) evaluation of the final retained solution, with a much larger sample size, say $M \gg N$. This evaluation gives us a more accurate estimate of the probabilities in the constraints of the original problem. Moreover, this evaluation is what really counts when we assess the quality of the solutions from our algorithms. After observing these probabilities in the out-of-sample simulation, we should increase the number of agents in the periods in which the constraints are violated, and we may decrease the number of staffs in the periods in which the estimates of the probabilities is much larger than the targets. After improving the staffing level, we can continue using another independent (out-of-sample) evaluation the model with the new staffings, by using a larger sample size, say $M' \gg M$. Based on this evaluation, we may continue improving the quality of the solution. This process can be repeated. Each time an out-of-sample simulation with a larger sample size is used, we try to improve the solutions, based on this evaluation, in order to get a staffing level which is closer to the optimal staffing. Obviously, we obtain a better solution with a larger sample size for the out-of-sample simulation.

CHAPTER 5

NUMERICAL EXPERIMENTS

In the previous chapter, we have proposed three algorithms to estimate the required staffing level for the chance-constrained staffing problem. In order to assess the performance of these algorithms, as well as the impact of flexibility on solutions, a number of models were fitted by our algorithms. These models were constructed to be representative of real data sets obtained from a 24-hour emergency call center (911).

5.1 Arrival process

In this section, we give definitions of several arrival processes, based on Oreshkin et al. [30].

We consider one day of operation of a call center. The opening hours are divided into P time periods of equal length. Let $\mathbb{X} = (X_1, \ldots, X_P)$ be the vector of arrival counts in those P periods. Assuming that the arrivals are from a Poisson process with a random rate Λ_p, constant over period p. Suppose $\Lambda = (\Lambda_1, \ldots, \Lambda_P)$ and $\Lambda_p = B_p \lambda_p$ where B_p is a non-negative random variable with $\mathbb{E}[B_p] = 1$ for each p. B_p is called the *busyness factor* for period p and denoting $\mathbb{B} = (B_1, \ldots, B_P)$. To summarize, we have:

$$\Lambda_p = B_p \lambda_p \text{ and } X_p \sim \mathit{Poisson}(\Lambda_p),$$

where $\mathit{Poisson}(\lambda)$ denotes the Poisson distribution with mean λ. Let $\Gamma(a,b)$ denote a gamma distribution with mean a/b and variance a/b^2. Here are several arrival processes we will consider in our models:

- The first case is the degenerate case where $B_p = 1$ for all p. It gives an ordinary nonhomogenous Poisson arrival process with piecewise constant rate. We will refer to this case simply as `PWCP`.
- In the second special case, one takes $B_p = B$ for all p, suppose B has a gamma distribution $\Gamma(\gamma, \gamma)$. We call this model `PWCPB`.

– The third setting uses independent busyness factors B_p for the different periods of the day. Suppose B_p has a gamma distribution $\Gamma(\rho_p, \rho_p)$. We refer to this model as PG.
– In the fourth case, we consider the following two-level arrival process model, named PGB, based on the multiplicative combination of independent period busyness factors $\widehat{B}_p$ and the busyness factor for the day, $\overline{B}$. We assume that $\overline{B}$, $\widehat{B}_1, \ldots,$ $\widehat{B}_P$ are independent with

$$\overline{B} \sim \Gamma(\beta, \beta) \text{ and } \widehat{B}_p \sim \Gamma(\alpha_p, \alpha_p) \text{ for each } p,$$

for some positive parameters β, $\alpha_1, \ldots,$ α_P, and we take

$$B_p = \widehat{B}_p \overline{B}$$

as the busyness factor of period p.
– The last case we consider is denoted PGNR. It is based on a normal copula for the vector $B = (B_1, \ldots, B_P)$. More specifically, each B_p is assumed to have a $\Gamma(\alpha_p, \alpha_p)$ distribution, with cumulative distribution function (cdf) G_p, and can be expressed as $B_p = G_p^{-1}(\Phi(Z_p))$, where Φ is the standard normal cdf and $Z = (Z_1, \ldots, Z_P) \sim$ Normal$(0, R^Z)$, a normal vector with mean zero and covariance matrix R^Z.

5.2 An emergency call center

5.2.1 Data from an emergency call center

The emergency call center operates 24 hours a day for 7 days a week. There is one skill group. The center receives calls categorized in several dozen types. For the results reported here, a subset of those types for which the daily patterns were similar is selected and the aggregated arrival process for those types is considered. These results are representative of a larger set of statistical analyses performed over different subsets and over individual call types having sufficiently large volume. The days are divided into P = 48 half-hour periods. Suppose that the callers do not abandon and the service

time is modelled using the *JohnsonSU* distribution. The models use the different arrival processes defined above. The numerical examples considered here are models obtained from real data sets from the 911 call center in Montreal, on Monday and Thursday. We shall denote each model by the name of the arrival process after the name of the day, e.g., `MondayPWCP` denotes the model with nonhomogenous Poisson arrival process with piecewise constant rate, on Monday.

5.2.2 A call center with very low occupancy

5.2.2.1 Parameters

The emergency call center 911 in Montreal requires that the SL must be very high and the average waiting times are very low, i.e., the agents have a low occupancy. The acceptable waiting time is **2 seconds**, and the target of SL is **0.95**. In our experiments in this section, we choose the parameters as follow: the acceptable waiting times in the SL constraints are $\tau_0 = \tau_p = 2$ seconds for all $1 \leq p \leq P$, the targets of service levels are $s_0 = s_p = 0.95$ for all p, the targets for the probabilities that the constraints on the SL are satisfied are $r_0 = 0.95$ and $r_p = 0.85$ for all p, the acceptable average waiting time in the AWT constraints are $w_0 = w_p = 2$ seconds, the targets for the probabilities that the constraints on the average waiting times are satisfied are $v_0 = 0.95$ and $v_p = 0.85$ for all p.

We next turn to experiment with the real data sets which we presented above. In order to evaluate the numerical potential of our three algorithms (CCS1, CCS2, CCS3) and compare them, we estimate the staffings for ten models by our algorithms in different cases. For each period, the initial staffing level is set to 0 or is given by the *Erlang C* formula. The computational times are observed to evaluate and compare the performance of the algorithms.

5.2.2.2 Comparisons of the three algorithms

We consider here our three algorithms with the initial staffings obtained by using the Erlang C formula, and by setting 0 for all periods. Note that in these cases, Erlang

C is used to find the staffing levels such that the proportion of calls answered within 2 seconds exceeds 95%.

Firstly, we consider the computational time of the three algorithms. We use these algorithms with the sample sizes 500, 1000 and 2000 for ten models. In all cases, the computational time of CCS1 are much worst, compared with the two other methods. The difference in the CPU times for CCS2 and CCS3 is not significant. The computational times of all three methods are better when using *Erlang C* to initialize the staffing level. Moreover, all the results show that when the sample size increases, the optimization time of each method also increases.

The three algorithms give very similar final solutions in all cases we tried. Figure 5.1 shows the detailed total costs when we optimize staffing level for ten models by using the three algorithms with 1000 replications, and the same sequence of random variables. The difference between results of the three algorithms could be explained that although the three algorithms are based on the same idea, but the methods to change the number of staffings and the order of periods in which we change the staffings in each algorithm are different. All of them have impacts on the final solutions. However, according to Figure 5.1, there are eight models which have the same results, and for the two models `MondayPGNR` and `ThursdayPGB`, the differences of the results of the three algorithms are only one agent (less than 0.23%) . Therefore, to assess the quality of solutions of our algorithms, we only consider method CCS3 with initial staffings obtained from the Erlang C formula.

5.2.2.3 Analysis of the staffing levels obtained from *Erlang C* and our algorithms

In each model in this case, Erlang C gives a staffing level which is less than the staffing level obtained from our method CCS3. Figure 5.2 shows the staffing level obtained by *Erlang C* and by our method CCS3 with the sample size 1000 for the model `MondayPGB`. According to our observations, although the Erlang C formula gives the staffings in all periods which are less or equal than the staffings obtained by CCS3, the distribution of the staffing level obtained from Erlang C is quite similar to the distribution of the staffing level obtained from our algorithms. For example, in both cases, the

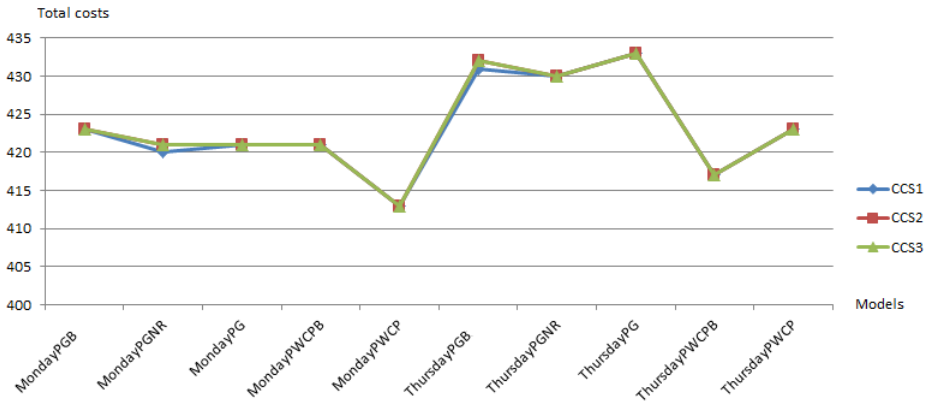

Figure 5.1: *Total costs of final solutions with the three algorithms for 1000 replications.*

numbers of agents in periods 9, 10, 11 are lowest, and the numbers of agents in periods 31, 32, 33 are highest. This can be explained by the different arrival rates of incoming calls between periods.

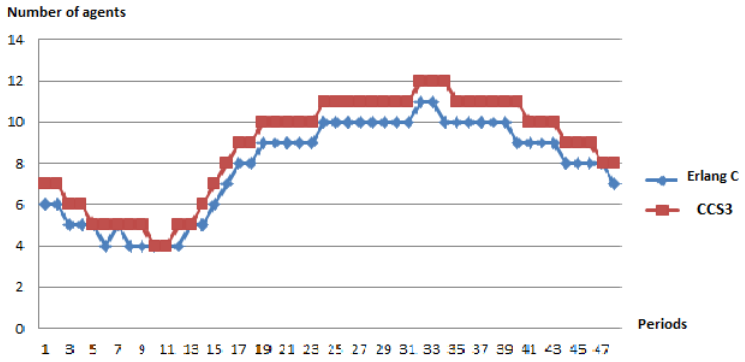

Figure 5.2: *Staffing levels obtained from Erlang C and algorithm* `CSS3` *for 1000 replications of* `MondayPGB`.

Next, we assess the efficiency of staffing levels obtained by our algorithms in comparison with using Erlang C formula. We use Erlang C to find the staffing levels such that the proportion of calls answered within 2 seconds is greater than 95%. After that, we evaluate this staffing level out-of-sample with a sample size 10000. Figure 5.3 shows the distribution of the service level in the whole day of the model `MondayPGNR` with the staffing level obtained by Erlang C. We also observe the proportion of days for which

the constraints on the service level and the average waiting time, for each period, are satisfied, over 10000 simulated days, in Figure 5.4 and Figure 5.5. The results show that for 87.5% of the periods, the SL constraints are violated, and in 10.4% of the periods the constraints on the AWT are not satisfied. The numerical result also shows that the probability that the service level over the whole day meets the demand is only 77.76%, while our target is 95%. In conclusion, in these models of 911, Erlang C formula gives staffing levels which are not good for our chance-constrained problem.

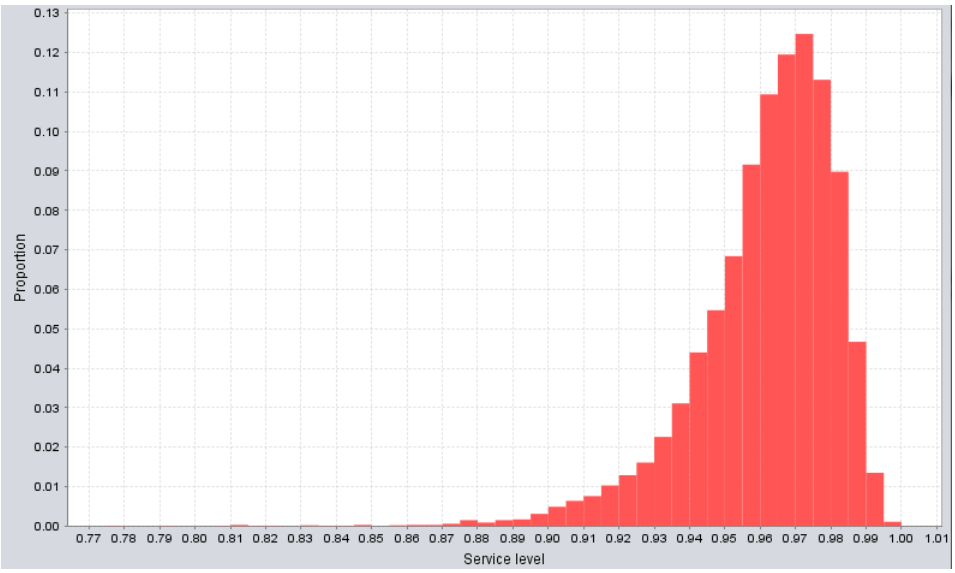

Figure 5.3: *The distribution of the SL in the whole day of the model* `MondayPGNR` *with the staffing level obtained by Erlang C.*

Next, we analyse the quality of solutions obtained by our algorithms. We use the method CCS3 to optimize the staffing level for the ten models with the sample size 1000. After that, we evaluate our solutions out-of-sample with the sample size 10000. Table 5.I shows the constraints which are not satisfied in the out-of-sample evaluations, for each model. According to these results, in all models we tried, for out-of-sample evaluations, the algorithm CCS3 gives us the staffing levels which satisfy most constraints. Moreover, when we consider the violated constraints, we see that the estimated probabilities in these constraints are very close to the target of 0.85. It means that CCS3 gives good results in these cases. Figure 5.6 shows the distribution of the service level in the whole day of the

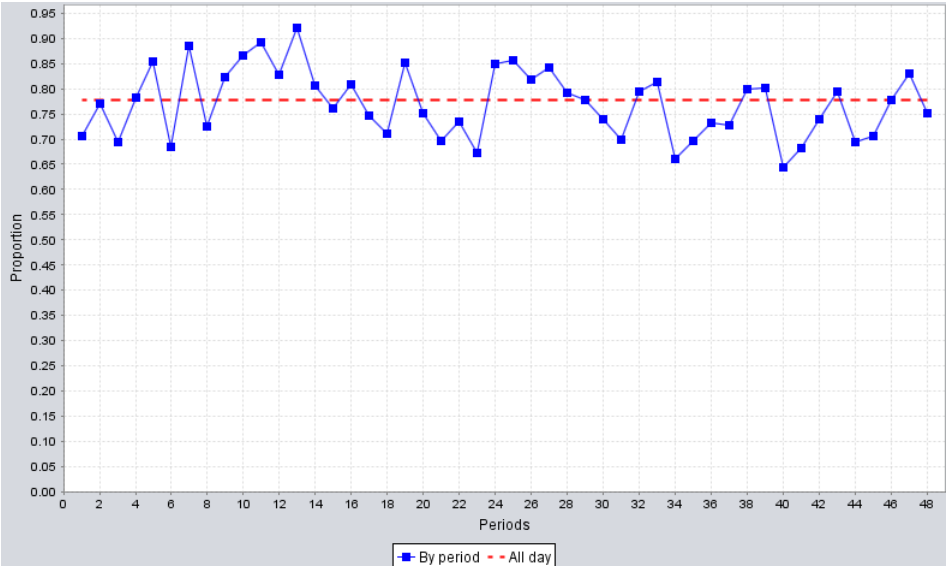

Figure 5.4: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by Erlang C.*

model `MondayPGNR`, with the staffing level obtained by CCS3. The numerical result shows that the probability that the service level over the whole day meets the demand is 99%, which exceeds our target of 95%. More specifically, we observe the proportion of the days where the constraints on the service level and average waiting time are satisfied, for each period, over 10000 simulated days, in Figure5.7 and Figure 5.8. According to these results, all chance constraints on the AWT and most constraints on the SL (except for the constraint in period 34) are satisfied. The estimated probability that the constraint on the SL in period 34 is satisfied, is 0.8419. This value is very close to the target of 0.85.

To improve the quality of these solutions, we try to increase the number of agents in periods that the constraints are not satisfied. We show an example of the model `MondayPGNR`. According to Figure 5.7, the SL constraint in period 34 is not satisfied. Then we try to increase the number of agents in period 34, from 11 to 12, and perform an out-of-sample evaluation for the model `MondayPGNR` with this new staffing level. According to Figure 5.9, the estimated probability that the SL constraint in period 34 is satisfied, is increased to 0.943, i.e. the chance constraint in SL in this period is satisfied.

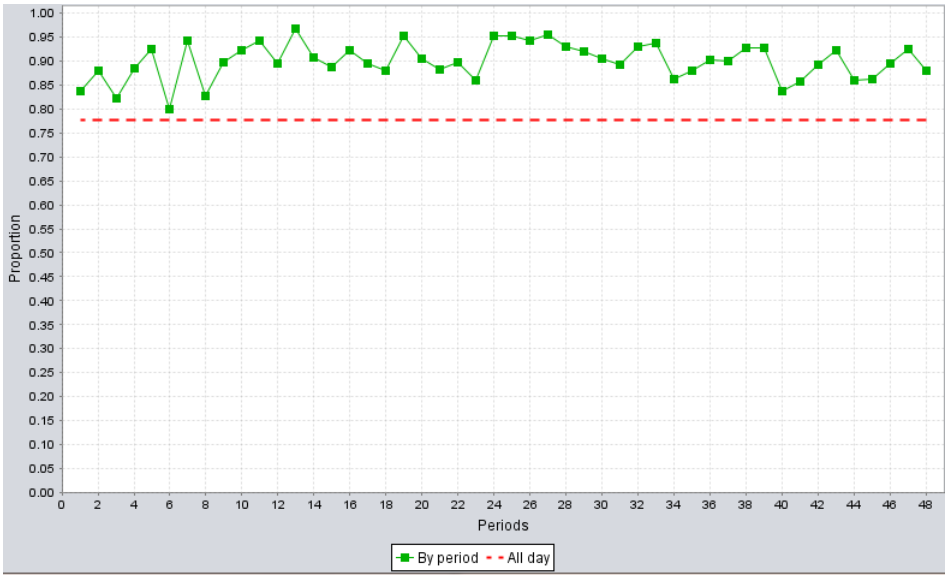

Figure 5.5: *Proportion of the days where the AWT constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by Erlang C.*

Now we try to improve the solutions by decreasing the number of agents. According to Figure 5.7, the estimated probability that the constraint on the SL in period 12 is satisfied is very large (0.9639), compared with the target of 0.85. We may expect that we can decrease the number of staffs in this period to still obtain a feasible solution for an out-of-sample evaluation. We try to decrease the number of agents in period 12, from 5 to 4, and perform an out-of-sample evaluation for the model `MondayPGNR` with this new staffing level. According to Figure 5.10, the estimated probability that the constraint on the SL in period 12 is satisfied is 0.8321, i.e., this staffing level is infeasible for the out-of-sample evaluation. We did this process for our ten models, and we conclude that in these cases, from the solutions obtained by the algorithm CCS3, we cannot decrease any agent in any period to obtain better results.

5.2.2.4 Analysis of the ten call center models

We remind that in our algorithms, simulation is used to estimate the service level and average waiting time for a given set of staffing levels. According to this, we can

Models	Violated constraints
MondayPGB	
MondayPGNR	$\mathbb{P}[S_{34}(2,y) \geq 0.95] = 0.8431$
MondayPG	$\mathbb{P}[S_5(2,y) \geq 0.95] = 0.8466$; $\mathbb{P}[S_{34}(2,y) \geq 0.95] = 0.8444$
MondayPWCPB	
MondayPWCP	$\mathbb{P}[S_{27}(2,y) \geq 0.95] = 0.8465$; $\mathbb{P}[S_{43}(2,y) \geq 0.95] = 0.8444$
ThursdayPGB	$\mathbb{P}[S_{24}(2,y) \geq 0.95] = 0.8315$
ThursdayPGNR	
ThursdayPG	
ThursdayPWCPB	$\mathbb{P}[S_{45}(2,y) \geq 0.95] = 0.8477$
ThursdayPWCP	$\mathbb{P}[S_{25}(2,y) \geq 0.95] = 0.8492$

Table 5.I: *Violated constraints for out-of-sample simulations of the ten models.*

compute the estimated probabilities that the SL and AWT constraints are satisfied. Each time running our algorithm to optimize staffing level for a model, a sequence of random variables is used to perform a simulation. In this part of the context, we evaluate the property of each model. We do it as follow: we use CCS3 to optimize staffing levels ten times with ten different sequences of random variables, for each model. The sample size is 1000. Table 5.II reports the total costs for ten simulation's times and the standard deviation of total staffing costs of ten models. According to our observation, we realize that the eight models of the arrival processes `PGB, PGNR, PG, PWCP` (on two different days) have very low standard deviations. It means that these models are reliable over different samples, i.e. using different sample sequences to optimize the staffing costs for these models, we obtain staffing levels with similar total costs. However, the models of the arrival process `PWCPB` on two different days have the highest standard deviations, about ten times higher than for other models. It shows the unreliable property in the solution of these models. The difference of the total costs obtained during the ten optimization runs, for each model, are very large. For example, for the model `MondayPWCPB`, for a sample sequence, we obtain the total cost of 370, while for another sample sequence, the total cost obtained is 421. The different between these costs are too large. We simulate this model with the staffing level with total cost 370, by using other sequence random variables, and observe that most chance constraints are violated.

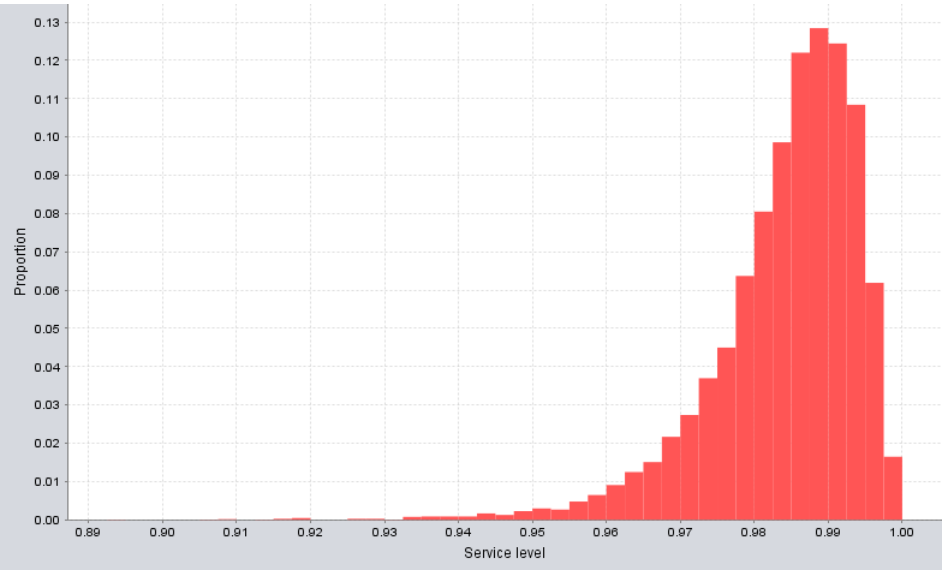

Figure 5.6: *The distribution of the service level in the whole day of the model* `MondayPGNR` *with the staffing level obtained by CCS3.*

There are 38 periods where the estimated probabilities that the constraints on the SL are less than the target of 0.85, a result not acceptable. The most plausible explanation is that, for a model with the arrival process `PWCPB`, there is a large variance in the call volumes. This model does not fit the variance equally well for all periods of a day (see Avramidis et al. [2] and Channouf and L'Ecuyer [13]). To obtain better results for these models, we should increase the sample size.

Sample	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}	Mean	Std (σ)
`MondayPGB`	423	422	422	423	422	422	419	426	421	420	422	1.7885
`MondayPGNR`	421	423	422	423	423	421	423	421	424	422	422.3	1.00499
`MondayPG`	421	420	421	422	421	422	420	421	422	419	420.9	0.9434
`MondayPWCPB`	421	370	415	414	399	417	387	420	427	402	407.2	16.8392
`MondayPWCP`	413	414	415	415	413	415	415	413	412	413	413.8	1.077
`ThursdayPGB`	432	428	433	430	434	430	430	431	428	429	430.5	1.9104
`ThursdayPGNR`	430	431	433	431	427	430	431	433	431	432	430.9	1.6401
`ThursdayPG`	433	429	431	429	429	431	429	428	430	432	430.1	1.5133
`ThursdayPWCPB`	417	421	404	427	426	464	389	399	412	424	418.3	19.3393
`ThursdayPWCP`	423	424	421	420	421	421	421	423	422	422	421.8	1.1662

Table 5.II: *Mean and standard deviation of the staffing costs with the sample size 1000.*

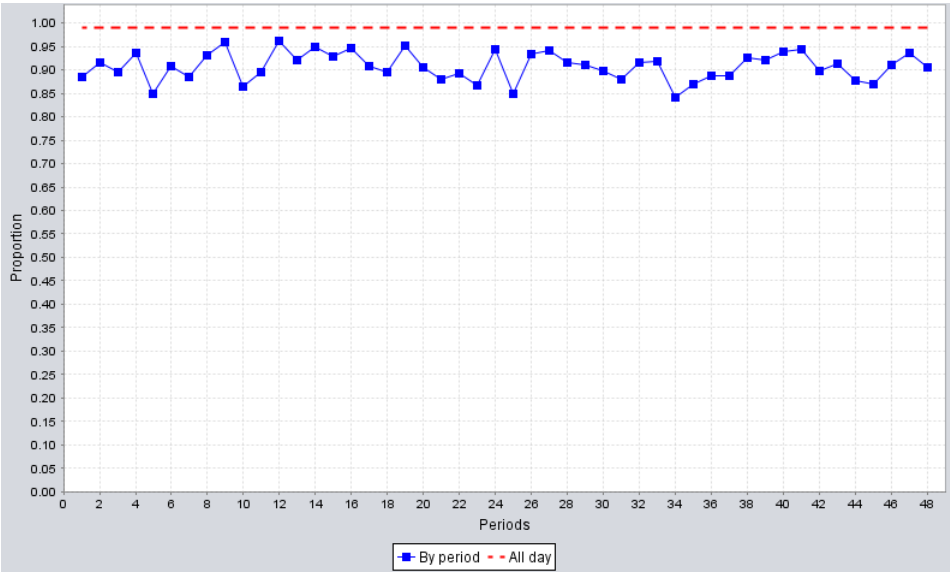

Figure 5.7: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by CCS3.*

5.2.3 A low occupancy call center

5.2.3.1 Parameters

In this section, we will assess the performance of our algorithms to optimize the staffing level in a call center where the QoS constraints are less demanding and the occupancy is higher compared with the call center in the previous section. The acceptable waiting time is **2 minutes**, and the target of the SL is **0.8**. More specifically, we choose the parameters as follow: the acceptable waiting times in SL constraints are $\tau_0 = \tau_p = 120$ seconds for all $1 \leq p \leq P$, the targets of service levels are $s_0 = s_p = 0.8$ for all p, the targets for the probabilities that the constraints on the SL are satisfied are $r_0 = 0.95$ and $r_p = 0.85$ for all p, the acceptable average waiting time in the AWT constraints are $w_0 = w_p = 120$ seconds, the targets for the probabilities that the constraints on the AWT are satisfied are $v_0 = 0.95$ and $v_p = 0.85$ for all p.

We still consider ten models which use different arrival processes as in the previous section, with some changes in targets of the SL and the acceptable waiting times. To avoid any confusion with the models in Section 5.2.2, we add the mark "*" with their

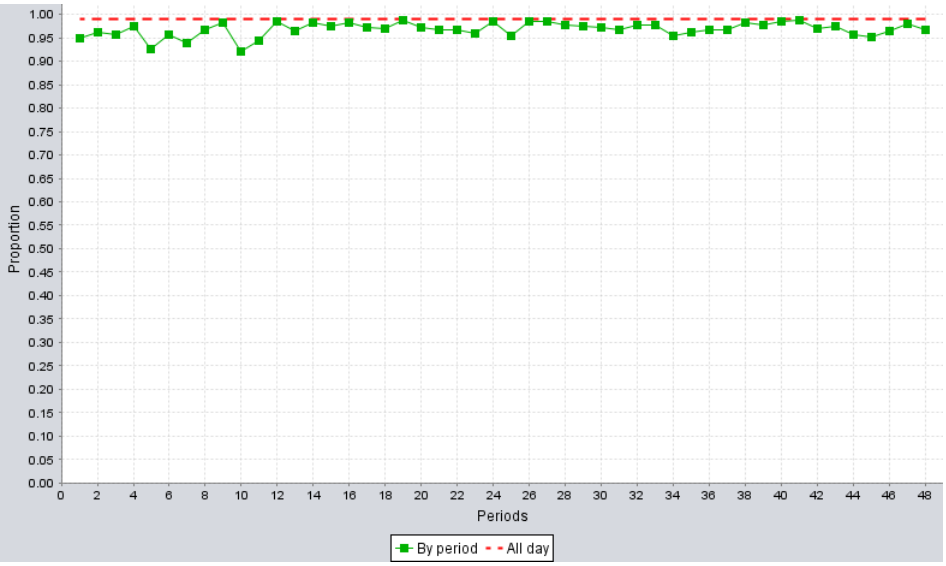

Figure 5.8: *Proportion of the days where the AWT constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by CCS3.*

names, e.g., `MondayPWCP`* denotes the model with nonhomogenous Poisson arrival process with piecewise constant rate, on Monday, in case the targets of service level are 0.8 and the acceptable waiting times are 120 seconds.

5.2.3.2 Analysis of staffing levels obtained from *Erlang C* and our algorithms

In all ten models we tried, the number of agents obtained by Erlang C is less or equal to the number of agents obtained by CCS3, in each period. However, the differences between these staffing levels obtained by Erlang C and by CCS3 are very small. Now we assess the quality of the solutions obtain from Erlang C and CCS3 by performing out-of-sample simulations with 10000 replications. We consider the model `MondayPGB`*. Figure 5.11 shows the staffing levels obtained by Erlang C and by CCS3 for `MondayPGB`*. Note that the sample size is 1000. These staffing levels are different by only one agent in period 8. Figure 5.12 shows the proportion of the days where the constraint in SL is satisfied,, for each period, over 10000 simulated days, of the model `MondayPGB`*, with the staffing level obtained by Erlang C. According to this result, the estimated probability that the constraint in SL is satisfied, in period 8, is only 0.8199, it is less than the target

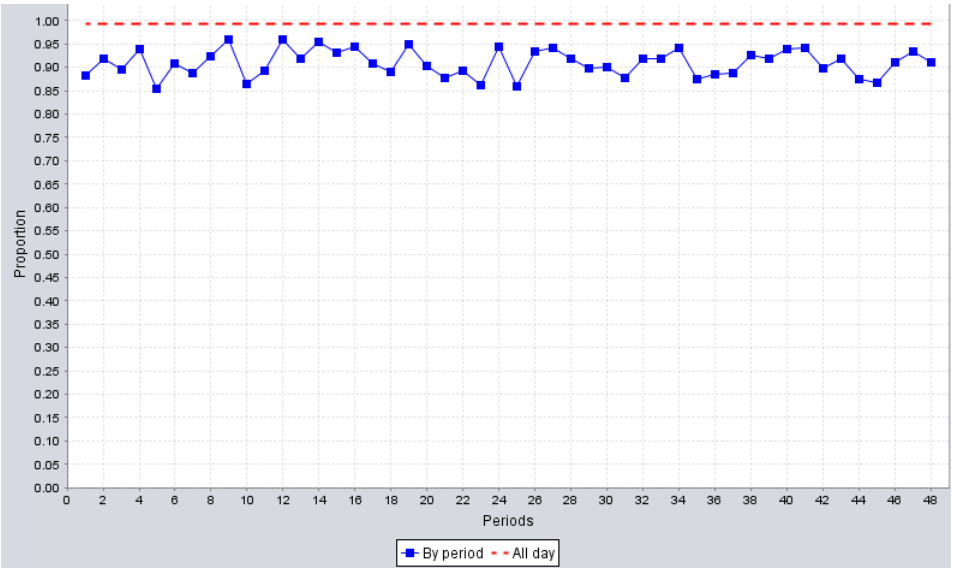

Figure 5.9: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGNR` *model, with the staffing level obtained by increasing one agent in period 34 from the staffing level obtained by CCS3.*

of 0.85. It means the staffing level obtained by Erlang C is infeasible for our chance-constrained problem. Figure 5.13 shows the estimated probability that the constraint on the SL is satisfied, for each period, of the model `MondayPGB`* with the staffing level obtained by CCS3. From our observation, the estimated probability that the constraint on the SL is satisfied in period 8, is much higher than the target of 0.85 (0.9917). As we observed in this Figure, this staffing level is feasible for our problem.

5.2.3.3 Analysis of solutions obtained by CCS3

In this section, we assess the quality of staffing levels obtained from CCS3 for the ten models. First, we obtain the solutions by using CCS3 to optimize staffings for our models with the sample size 1000. Then, we perform out-of-sample simulations with 10000 replications for these models with these new staffings. Table 5.III shows the constraints which are not satisfied in these out-of-sample simulations. There are eight models where all constraints are satisfied, and only two, `MondayPGNR`* and `ThursdayPWCP`*, where a violated constraint exists. However, the estimated probabil-

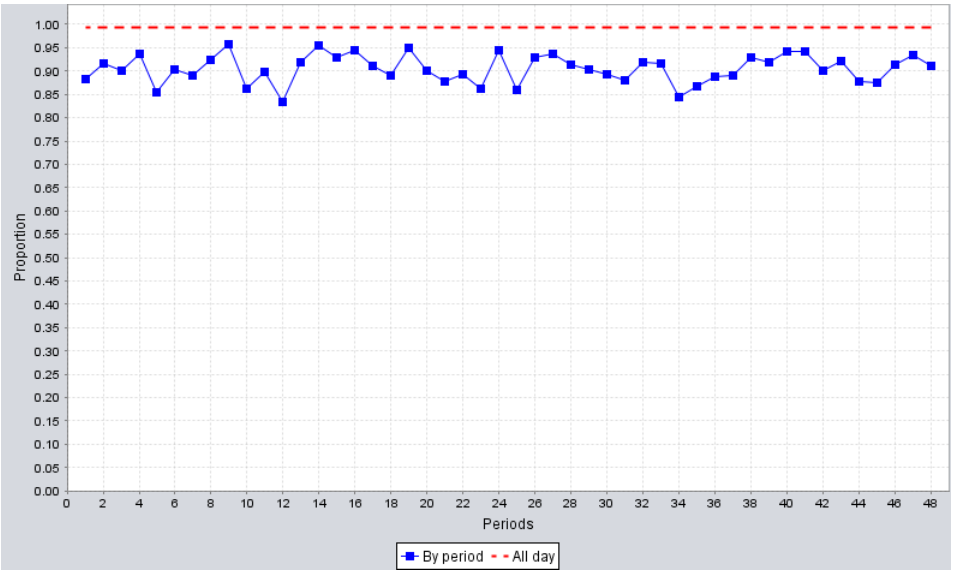

Figure 5.10: *Proportion of the days where the SL constraint was satisfied, for each period, over 10000 simulated days, for the* `MondayPGBNR` *model, with the staffing level obtained by decreasing one agent in period 12 from the staffing level obtained by CCS3.*

ities that the constraints on the SL are satisfied, in these constraints, are very close to our target of 0.85.

To improve these solutions, we try to increase the number of agents in these periods. Figure 5.14 shows the proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the `MondayPGNR`* model, with the staffing level obtained by CCS3, and Figure 5.15 shows the proportion of days where the SL constraint is satisfied, over 10000 simulated days, for the `MondayPGNR`* model, with the staffing level obtained by adding one agent in period 43 from the staffing level obtained from CCS3. According to this Figure, all the constraints are satisfied, the estimated probability that the constraint on the SL in period 43 is now 0.9736, much higher than 0.85.

We now continue trying to improve the solution of this model. According to Figure 5.14, the estimated probability that the constraint on the SL in period 8 is highest (0.9906). Thus, we try to decrease the number of agents in this period, from 3 to 2, and perform an out-of-sample evaluation with 10000 replications. Figure 5.16 shows the estimated probability that the constraint on the SL is satisfied, for each period, in this

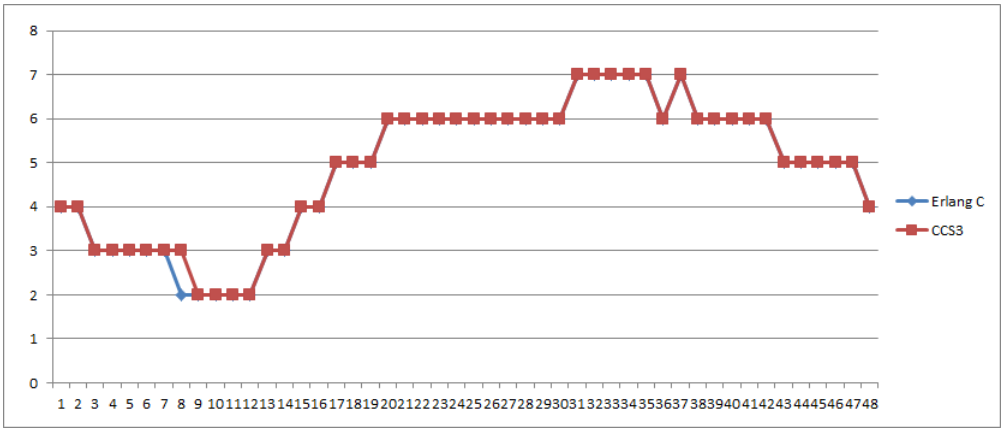

Figure 5.11: *Staffing levels obtained by Erlang C and CCS3 of the* `MondayPGB`* *model.*

case. As we saw in this Figure, the estimated probability that the constraint on the SL in period 8 is satisfied, is 0.8048. It is now less than the target of 0.85, i.e., the new staffing level is infeasible. Therefore, we cannot decrease any agent in this period. For each of ten models, after obtaining the staffing level by using CCS3, we try to decrease the number of staffing in each period, but these staffing levels are infeasible. Thus, we can expect that CCS3 gives good results in these cases.

Models	Violated constraints
`MondayPGB`*	
`MondayPGNR`*	$\mathbb{P}[S_{43}(120,y) \geq 0.8] = 0.848$
`MondayPG`*	
`MondayPWCPB`*	
`MondayPWCP`*	
`ThursdayPGB`*	
`ThursdayPGNR`*	
`ThursdayPG`*	
`ThursdayPWCPB`*	
`ThursdayPWCP`*	$\mathbb{P}[S_9(120,y) \geq 0.8] = 0.8412$

Table 5.III: *The violated constraints for the out-of-sample simulation of the ten models with low occupancy.*

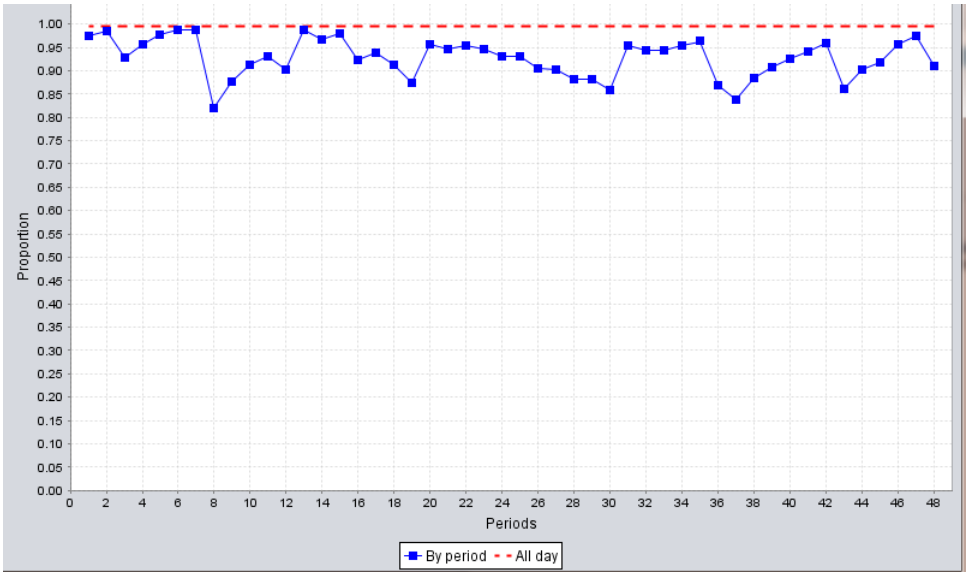

Figure 5.12: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGB`* *model with the staffing level obtained by Erlang C.*

5.3 A call center with high occupancy

5.3.1 Parameters

In this section, we consider a call center which has higher occupancy. The acceptable waiting time now is **5 minutes**, and the target of the SL is **0.8**. More specifically, we choose the parameters as follow: the acceptable waiting times in the SL constraints are $\tau_0 = \tau_p = 300$ seconds for all $1 \leq p \leq P$, the targets of service levels are $s_0 = s_p = 0.80$ for all p, the targets of the probabilities that the constraints on the SL are satisfied are $r_0 = 0.95$ and $r_p = 0.85$ for all p, the acceptable average waiting time in the AWT constraints are $w_0 = w_p = 300$ seconds, the targets of the probabilities that the constraints on AWT are satisfied are $v_0 = 0.95$ and $v_p = 0.85$ for all p. We still consider ten models, using the different arrival processes in the previous sections, with some changes in targets of SL and acceptable waiting times. To avoid confusion with the models in Section 5.2.2 and Section 5.2.3, we add the mark "**" with their names, e.g. `MondayPWCP`** denotes the model with nonhomogenous Poisson arrival process with piecewise constant rate, on Monday, in case the targets of the SL are 0.8 and the acceptable waiting times are 300

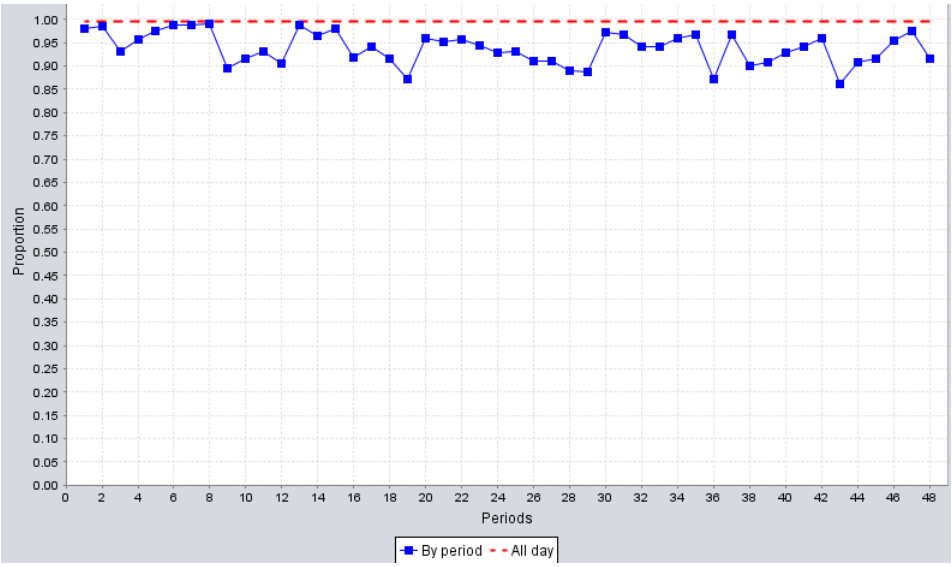

Figure 5.13: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* `MondayPGB`* *model with the staffing level obtained by CCS3.*

seconds.

5.3.2 Analysis staffing levels obtained by Erlang C and method CCS3

As we discussed in two previous sections, the number of agents in each period obtained by Erlang C is less or equal the number of agents obtained by CCS3. Nevertheless, in this case with higher acceptable waiting time, the staffing levels obtained by Erlang C are greater or equal to the staffing levels obtained by CCS3. Figure 5.17 shows the staffings obtained by Erlang C and by CCS3 for `MondayPGB`**.

5.3.3 Analysis of the solutions of our algorithms

We use CCS3 to optimize the staffing for our ten models with the sample size 1000. In order to assess the quality of these solutions, we perform out-of-sample simulations with 10000 replications with the found staffings. In each model, all constraints are satisfied in the out-of-sample evaluation. For all the models considered in the two previous sections, we realize that the stage `Correction` does not change the staffing in any

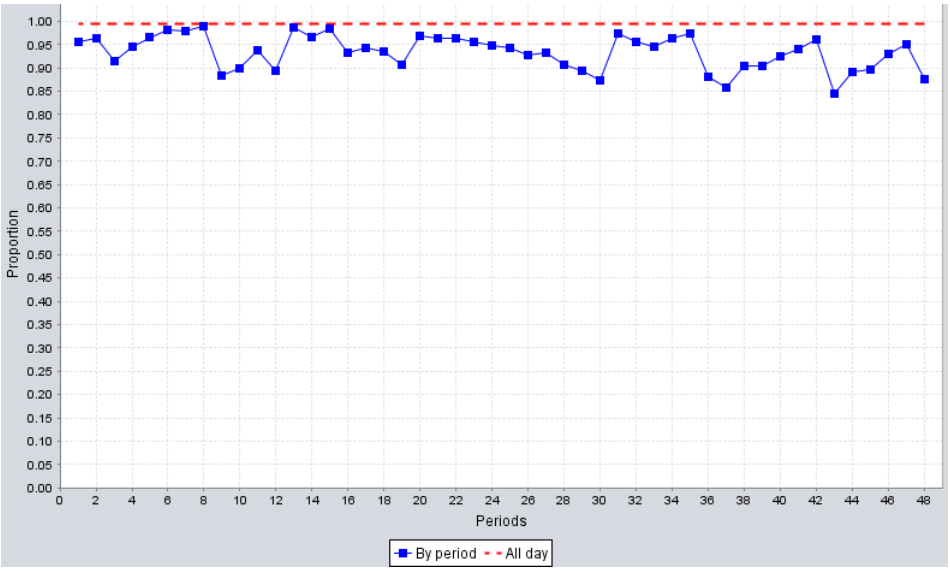

Figure 5.14: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGNR`* *model, with the staffing level obtained from CCS3.*

period. However, for the models in this section, the stage `Correction` helps reducing the number of agents in some periods. Figure 5.18 shows the staffing levels obtained by CCS3 before and after adding the new stage `Correction`. According to this Figure, the number of agents in periods 3, 14, 16, 42, 48 are decreased. Now we explain the reason why the stage `Correction` changes the result in this case. As we discussed in Section 4.1, changing the staffing in one period can change the performance (such as service level, etc.) in other periods as well. In the case of an emergency call center, since it has low occupancy (the acceptable waiting time is only 2 seconds or 2 minutes), there is rarely a queue in the system (the agents are not very busy), so this effect is very small. However, in this case, the occupancy is high (the acceptable waiting time is 300 seconds), the dependence between periods is larger. So, when we decrease or increase the staffings in a period, it has impact on the performance in other periods. We remind that in our algorithm CCS3, we change (increase and decrease) the number of agents in several periods at the same time, i.e., *in each iteration, we will increase or decrease the number of agents in all selected periods (where increase is required or decrease appears to be acceptable) simultaneously*. Now we show a specific example. Suppose that in

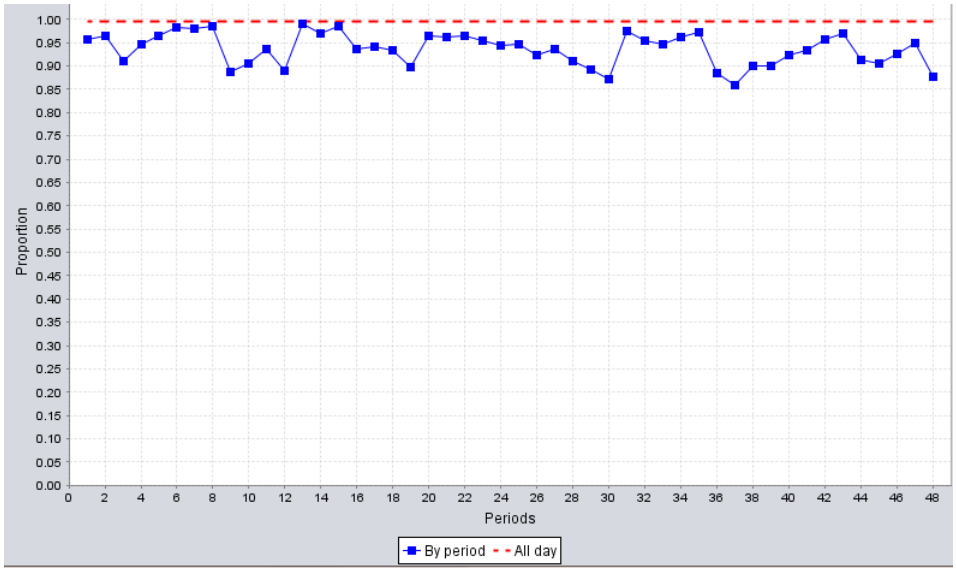

Figure 5.15: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGNR`* *model, with the staffing level obtained by increasing one agent in period 43 from the staffing level obtained by CCS3.*

the current iteration, we obtain a staffing level which does not satisfy the constraints in SL in periods 2 and 3. Then, in the next iteration, the number of agents in both these periods are increased. However, when increasing the staffings in period 2, not only the constraint in period 2, but also the constraint in period 3, is satisfied. However, CCS3 still increases the staffs in period 3, possibly overestimating the staffing levels.

In order to assess the quality of the result for the model `MondayPGB`**, we perform an out-of-sample evaluation with 10000 replications. Figure 5.19 shows the proportion of the days where the SL constraint in each period was satisfied, over 10000 simulated days, for the `MondayPGB`** model, with this new staffing level. According to this Figure, the estimated probabilities that the SL constraints are satisfied in all periods are higher than the target of 0.85. As we observed in Figure 5.19, the estimated probability in period 11 is still very high (0.998). We may expect that we can get a better result, by decreasing the staffs in period 11. However, after we reduce one agent in period 11, the estimated probability that the SL constraint in this period is satisfied, is 0.7559 (see Figure 5.20), less than the target of 0.85. We did the same process for all our models, and we conclude that, in all models we tried, by using our algorithms, we get good results,

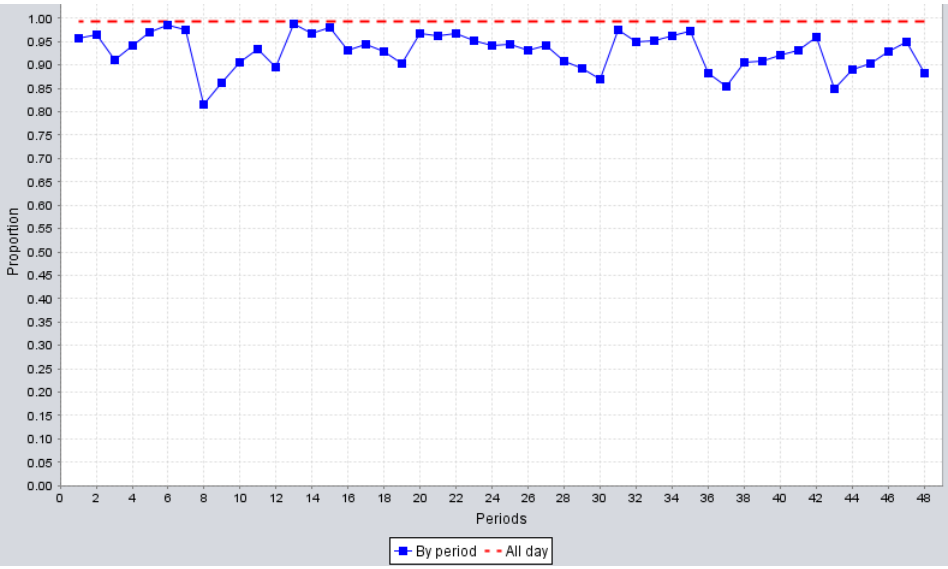

Figure 5.16: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGBNR`* *model, with the staffing level obtained by decreasing one agent in period 8 from the staffing level obtained by CCS3.*

and we cannot decrease any agent in any period to obtain better results.

5.4 A call center with larger arrival rate

5.4.1 Parameters

In the previous sections, we have assessed the performance of our algorithms by using data from an emergency call center 911 in Montreal, in different cases with the occupancy is very low (the AWT is 2 seconds), low (the AWT is 2 minutes) and high (the AWT is 5 minutes). When performing out-of-sample simulations these models using the staffings obtained by our algorithms, with the much higher sample sizes, most constraints are satisfied. The difference between the estimated probabilities and the targets, in all violated constraints, are very small. Moreover, we cannot improve these staffings by decreasing any agent in any period. Therefore, the three algorithms give good results in these cases. All these models used in the previous sections have a low traffic. In this section, we would like to assess the quality of the three algorithms in a call center with a higher traffic. The arrival rate in each period is ten times higher, compared to the call

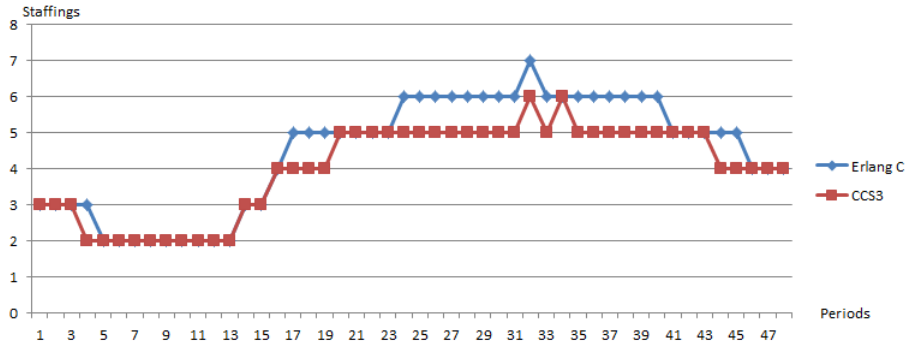

Figure 5.17: *Staffing levels obtained by Erlang C and CCS3 of the* `MondayPGB`** *model.*

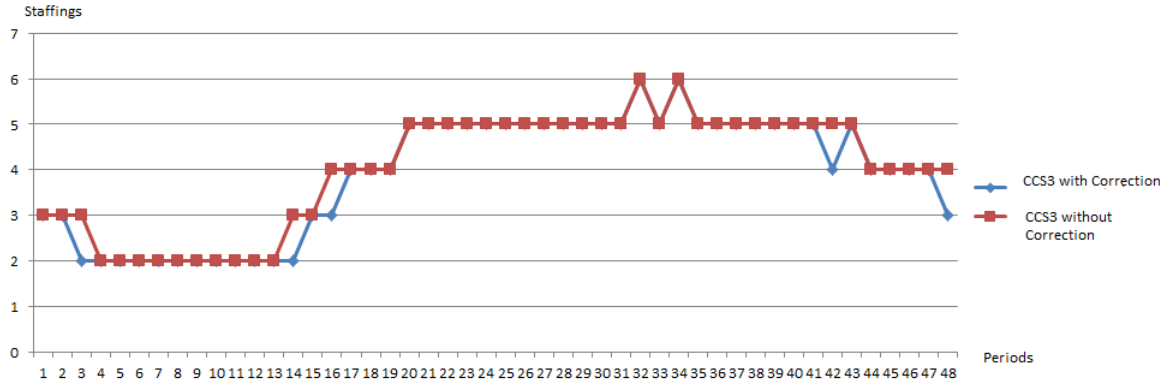

Figure 5.18: *Staffing levels obtained from CCS3 before and after adding stage* `Correction` *of* `MondayPGB`**.

center in 911. We consider eight models which use different arrival processes (`PGB`, `PG`, `PGNR`, `PWCP`), on Monday and Thursday. The acceptable waiting time is **2 minutes**, and the target of the SL is **0.8**. More specifically, the acceptable waiting times in the SL constraints are $\tau_0 = \tau_p = 120$ seconds for all $1 \leq p \leq P$, the targets of service levels are $s_0 = s_p = 0.8$ for all p, the targets for the probabilities that the constraints on the SL are satisfied are $r_0 = 0.95$ and $r_p = 0.85$ for all p, the acceptable average waiting time in the AWT constraints is $w_0 = w_p = 120$ seconds for all p, the targets for the probabilities that the constraints on the AWT are satisfied are $v_0 = 0.95$ and $v_p = 0.85$ for all p.

To avoid confusion with the models in Section 5.2.2, we add the mark "2" with their

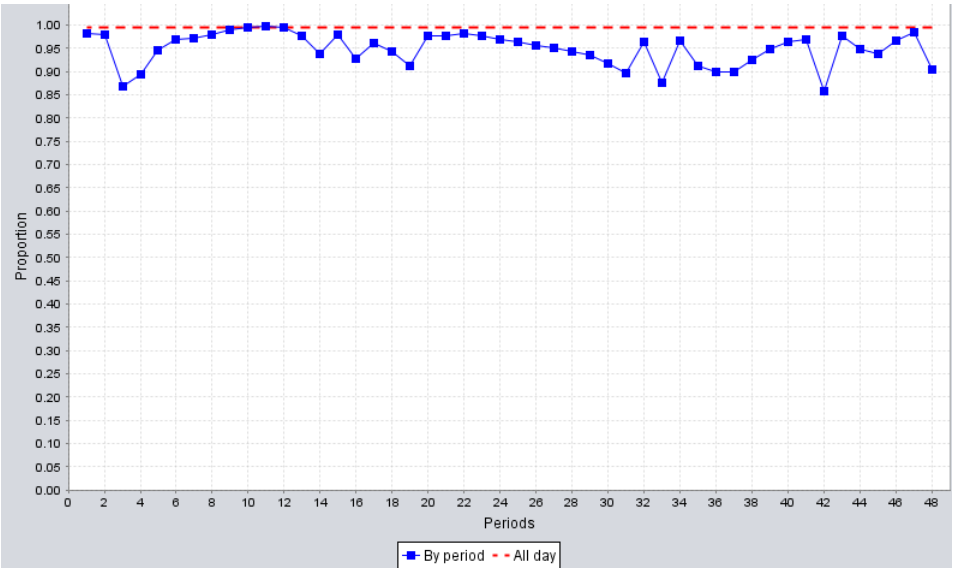

Figure 5.19: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* $\texttt{MondayPGB}^{**}$ *model, with the staffing level obtained by CCS3.*

names, e.g., $\texttt{MondayPWCP}_2$ denotes the model with nonhomogenous Poisson arrival process with piecewise constant rate, on Monday, in case the arrival rate in each period is ten time higher, compared with the call center 911, the targets of the SL are 0.8 and the acceptable waiting times are 120 seconds.

5.4.2 Analysis of staffings obtained by Erlang C and CCS3

In each model we tried, Erlang C gives a staffing level which is less than the staffing level obtained from our method CCS3. Figure 5.21 shows the staffing level obtained by *Erlang C* and by our method CCS3 with the sample size 1000, for the model $\texttt{MondayPGB}_2$. According to our observations, the differences between the staffings obtained by Erlang C and CCS3 are very small. Therefore, when we use the three algorithms to optimize staffing levels, initializing the staffings by using the Erlang C formula will help reducing the CPU time.

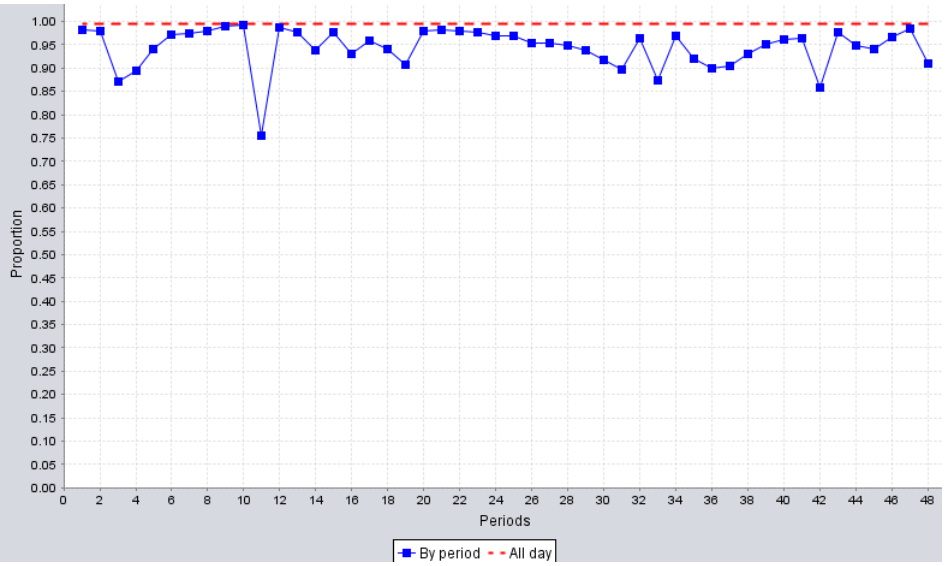

Figure 5.20: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* `MondayPGB`** *model, with the staffing level obtained by decreasing one agent in period 11 from the staffing level obtained by CCS3.*

5.4.3 Analysis of the solutions of our algorithms

In this section, we assess the quality of staffing levels obtained from CCS3 for the eight models. Fist, we obtain the solutions by using CCS3 to optimize staffings for our eight models with the sample size 1000. Then, we perform out-of-sample simulations with 10000 replications for these models with the new staffings. Table 5.IV shows the constraints which are not satisfied in these out-of-sample evaluations.

As we observed, for all models we tried, there are a few violated constraints. However, the estimated probabilities in these constraints are very close to the targets of 0.95 for the whole day and 0.85 for each period. To improve the quality of these solutions, we increase the staffing in the periods in which a constraint is not satisfied. Now, we try to improve the solutions, by decreasing the number of agents in some periods. Figure 5.22 shows the proportion of the days where the SL constraint is satisfied, and Figure 5.23 shows the distribution of the SL in the whole day, over 10000 simulated days, for the `MondayPWCP`$_2$ model, with the staffing level obtained by CCS3. According to Figure 5.22, the estimated probability that the constraint in SL in period 46 is very high

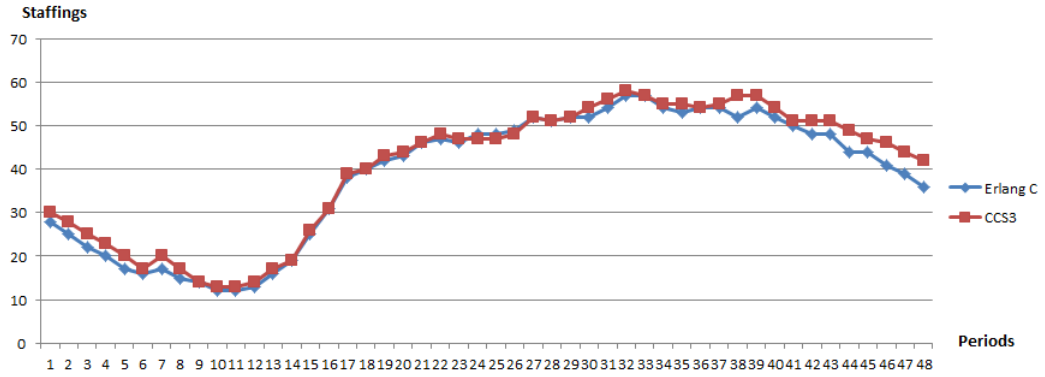

Figure 5.21: *Staffing levels obtained from Erlang C and algorithms for 1000 replications of* $\texttt{MondayPGB}_2$.

(0.9545). Thus, we try to decrease the number of agents in this period, from 35 to 34, and perform an out-of-sample evaluation. Figure 5.24 shows the estimated probability that the constraint on the SL is satisfied, for each period, in this case. As we saw in this Figure, the estimated probabilities that the constraint in SL in period 46 and 47 are satisfied, are decreased. The constraint on the SL in period 46 is still satisfied, the estimated probability is now decreased to 0.91. However, the estimated probability that the constraint on the SL in period 47 is now 0.8355, i.e., this constraint is not satisfied. When the staffings in period 46 is decreased, many arrival calls in this period are not served, and they have to wait in a queue, so the waiting in the next period tends to increase. It means that the service level in period 47 depends on the staffing level in the previous period. Thus, the estimated probability that the constraint on the SL in period 47 is decreased to less than the target 0.85. In the examples which we consider in the previous sections, when we decrease the staffings in a period from the staffings obtained by our algorithms, then this does not effect to another period much. However, in this case, when we decrease the number of agents in a period, the performance in other periods are effected as well. When the arrival rate in each period is high, the dependence between periods is large. In the eight models considered in this section, from the results obtained by CCS3, when we decrease the number of staffs in a period, the performance in other periods are changed as well, and the constraints in some periods can be violated.

Models	Violated constraints
MondayPGB$_2$	$\mathbb{P}[S_0(120,y) \geq 0.8] = 0.9453$
MondayPGNR$_2$	
MondayPG$_2$	$\mathbb{P}[S_{47}(120,y) \geq 0.8] = 0.8451$
MondayPWCP$_2$	$\mathbb{P}[S_{29}(120,y) \geq 0.8] = 0.8430$; $\mathbb{P}[S_{30}(120,y) \geq 0.8] = 0.8432$; $\mathbb{P}[S_{34}(120,y) \geq 0.8] = 0.8422$
ThursdayPGB$_2$	$\mathbb{P}[S_0(120,y) \geq 0.8] = 0.92$
ThursdayPGNR$_2$	$\mathbb{P}[S_0(120,y) \geq 0.8] = 0.9319$; $\mathbb{P}[S_9(120,y) \geq 0.8] = 0.8432$
ThursdayPG$_2$	$\mathbb{P}[S_{46}(120,y) \geq 0.8] = 0.8475$
ThursdayPWCP$_2$	$\mathbb{P}[S_{19}(120,y) \geq 0.8] = 0.8455$; $\mathbb{P}[S_{25}(120,y) \geq 0.8] = 0.8468$

Table 5.IV: *The constraints which are not satisfied for the out-of-sample simulation of the eight models.*

5.5 Summary

In this section, we have analysed the performance of our three algorithms in several cases: for an emergency call center with very low occupancy, low occupancy and high occupancy; and for a call center with higher arrival rates. We compare the quality of solutions obtained by our algorithms and by the Erlang C formula. In all cases in our experiment, our algorithms always give much better results than Erlang C. Moreover, we assess the quality of solutions of our algorithms by perform out-of-sample evaluations. According to these results, we try to improve the quality of solutions. We also analyse the dependence between periods in the models with high occupancy and the models with heavy traffic. In conclusion, our algorithms give good results in the models we tried.

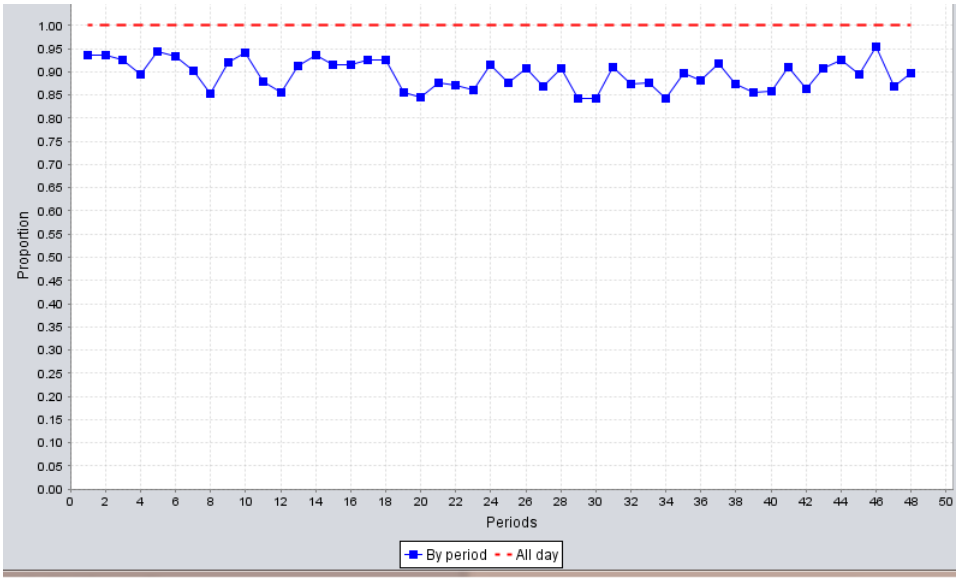

Figure 5.22: *Proportion of the days where the SL constraint is satisfied, for each period, over 10000 simulated days, for the* $\texttt{MondayPWCP}_2$ *model, with the staffing level obtained by CCS3.*

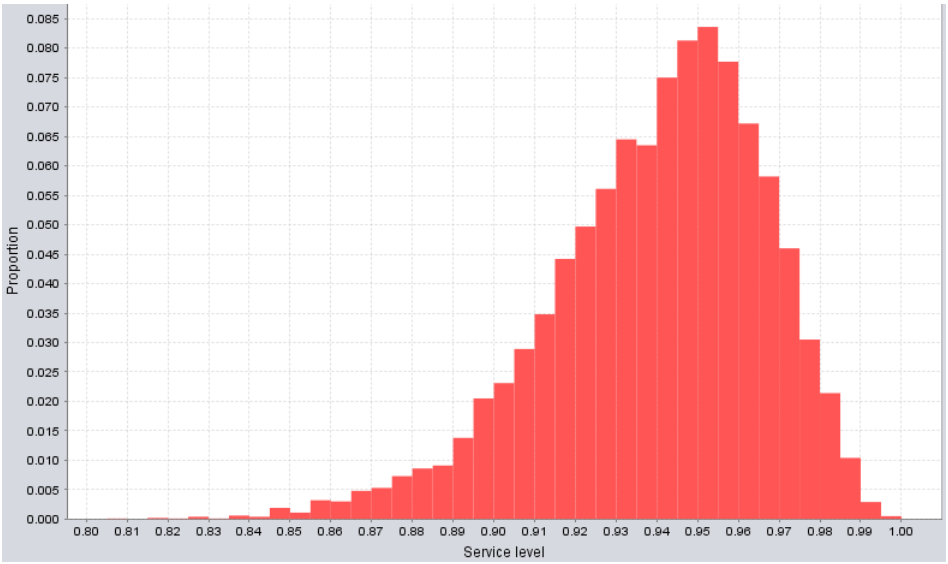

Figure 5.23: *The distribution of the service level in the whole day of the model* $\texttt{MondayPWCP}_2$ *with the staffing level obtained by CCS3.*

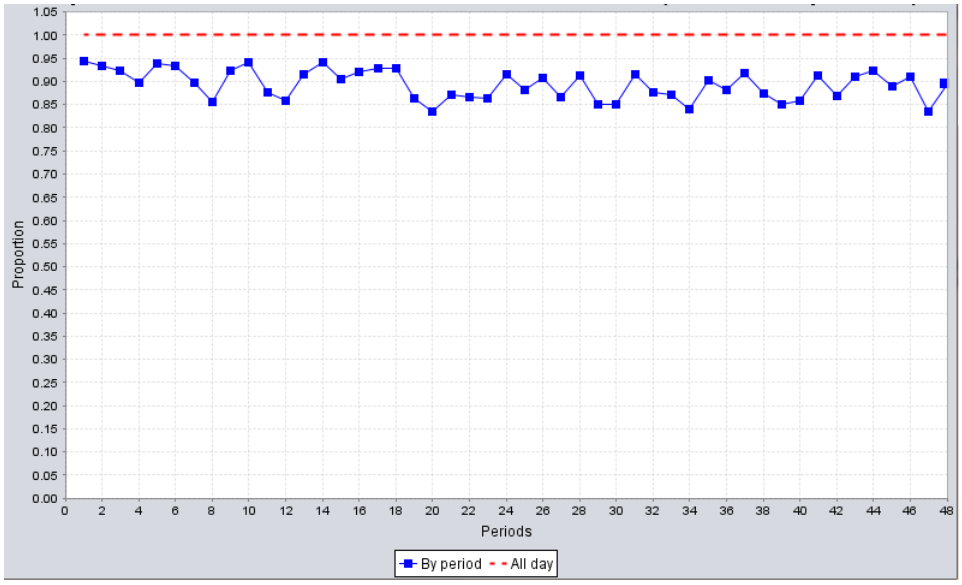

Figure 5.24: *Proportion of the days where the SL constraint is satisfied, over 10000 simulated days, for the* $\texttt{MondayPWCP}_2$ *model, with the staffing level obtained by decreasing one agent in period 46 from the staffing level obtained by CCS3.*

CHAPTER 6

CONCLUSIONS AND FURTHER RESEARCH PERSPECTIVES

In this final chapter, we will review the research contributions of this dissertation and discuss directions for future research.

6.1 Conclusions

In this thesis, we have modelled the chance-constrained staffing problem which requires that the QoS constraints are met with high probability with respect to the uncertainty in the demand rate. We are interested in constraints on the service level and the daily average waiting time. The QoS estimates are based on simulation with time-varying and stochastic arrivals rate. We define the sample average approximation of this problem and studied the convergence of the optimal solution of the sample problem to that of the original problem. We show that we can expect to get an optimal solution for the original problem by solving the sample problem if we choose a large sample size. Moreover, we show that the probability that the optimal solution of the sample problem is an optimal solution of the original problem, approaches 1 exponentially fast as we increase the sample size. The method of combining simulation and optimization has potential applications in solving the staffing problem. In this thesis, we propose three simulation-based optimization algorithms to solve our chance-constrained staffing problem with multiple call types, one agent group and multiple periods. All of these have the same idea with some variations in the method to change the number of agents in each period. In this context, we introduce two ways to initialize the staffing levels: by using the Erlang C formula, and by setting the staff number equal to 0 for all periods. In algorithm CCS1, we increase or decrease the number of agents in a single period at most one unit at each iteration, while in method CCS2, in each iteration, the number of agents are increased or decreased by at most one unit, but in all required periods simultaneously. To improve this method, we use bisection to increase or decrease the number of staffing

in algorithm CCS3.

We made several observations about the performance of these methods when applying them in different situations. First, we consider call center models with different arrival processes in case of low occupancy. These models are obtained from real data of an emergency call center 911 in Montreal. We compare the performance of the three algorithms in several cases: initial staffing obtained by using the *Erlang C* formula and initial staffing equal to 0 for all periods. In each situation, the difference of the solutions of these methods is very small if we use the same sequence of random draws. The CPU time of the three algorithms are reduced when we use Erlang C to initialize the staffings. However, the computational time of these algorithms have large difference. Method CCS1 is always the slowest in all cases we tried. To assess the quality of the solutions, we perform out-of-sample evaluations with much larger sample sizes. In this case, our methods give good staffing levels which satisfy most constraints, and we cannot decrease any agent in any period to get better results. The quality of the found solution tends to be better with larger sample sizes. We also analysed our call center models by using our algorithms. We conclude that the call center models which uses a Poisson process with piecewise constant rate approximation with common daily business factor (`PWCPB`) give large variances in the call volume, so the solutions to optimize staffings in these models have large standard deviations. To investigate the quality of the methods in more different cases, we consider some call center models where the occupancy is higher and the QoS constraints are less demanding. In the case where the acceptable waiting time is 2 minutes or 5 minutes, our methods still give good results. In all models, the solutions satisfy most constraints. For the models with acceptable waiting times of 2 seconds or 2 minutes, we observe that the last stage `Correction` does not change agents in any period. On the other hand, when we increase the acceptable time to 5 minutes, our algorithms without the stage `Correction` overestimates the staffing levels. It can be explained by the fact that in these models, the SL and AWT in one period can depend on the number of agents in other periods, either before or after. The stage `Correction` helps improving the quality of the solutions. This stage helps reducing the agents in several periods under the condition that all constraints are still satisfied. After adding

this stage, our algorithms give good results in all models we tried, i.e., in the out-of-sample evaluation, most constraints are satisfied, and we cannot decrease any agent in any period to obtain better results.

After that, we consider another call center in which the arrival rate in each period is ten time higher, compared with the call center 911. When we perform out-of-sample simulations, in all models in this case, most constraints are satisfied. Then, we try to decrease the staffings in a period, and observe the the SL changes not only for this period, but also for contiguous periods. The estimated probability on the SL in some periods decrease to less than the target. This can be explained that in a call center with heavy traffic of arrival calls, the dependence of performance between periods is large. In conclusion, our algorithms give good results in these models we tried.

6.2 Further research perspectives

In order to extend the idea in this thesis, we will continue investigating the staffing and schedule problem in call centers with chance constraints. As far as we discussed in this thesis, although the three algorithms we proposed are very easy to implement and they also give good results in some situations we tried, they are used only for a special case where all agents can answered all types of calls. Therefore, we would like to apply other methods to solve the chance constraints staffing problem. One of the methods will be investigated is the *cutting plane method* which we discussed in Section 4.6. We may expect that by using this method, we can also solve the chance-constrained staffing and schedule problems for multiskill call centers. We hope that we can find the difficulties encountered with those problems, and expect to develop methods to deal with these problems.

BIBLIOGRAPHY

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

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

[3] A. N. Avramidis, W. Chan, and P. L'Ecuyer. Staffing multi-skill call centers via search methods and a performance approximation. IIE Transactions, 41:483–497, 2009.

[4] A. N. Avramidis, W. Chan, M. Gendreau, P. L'Ecuyer, and O. Pisacane. Optimizing daily agent scheduling in a multiskill call centers. European Journal of Operational Research, 200(3):822–832, 2010.

[5] S. Bhulai, G. Koole, and A. Pot. Simple methods for shift scheduling in multi-skill call centers. Manufacturing and Service Operations Management, 10:411–420, 2008.

[6] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. Journal of the American Statistical Association, 100:36–50, 2005.

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

[8] Bureau of Labor Statistics. Occupational Outlook Handbook - Customer Service Representatives, 2010-11 Edition. 2007. U.S. Department of Labor. Available online at `http://www.bls.gov/oco/ocos280.htm`, (last accessed September, 2011).

[9] Bureau of Labor Statistics. Occupational employment and wages, 2007. September 2010. U.S. Department of Labor. Available online at http://www.bls.gov/oes/2007/may/oes_nat.htm, (last accessed September, 2011).

[10] Bureau of Labor Statistics. Occupational employment and wages, May 2010 - Customer Service Representatives. May 2011. U.S. Department of Labor. Available online at http://www.bls.gov/oes/current/oes434051.htm, (last accessed September, 2011).

[11] Bureau of Labor Statistics. An overview of U.S. occupational employment and wages in 2010. July 2011. U.S. Department of Labor. Available online at http://www.bls.gov/oes/highlight_2010.pdf, (last accessed September, 2011).

[12] M. T. Ceẓik and P. L'Ecuyer. Staffing multiskill call centers via linear programming and simulation. Management Science, 54(2):310–323, 2008.

[13] N. Channouf and P. L'Ecuyer. A normal copula model for the arrival process in a call center. International Transactions in Operational Research, 19:771–787, 2012.

[14] R. B. Cooper. Introduction to Queueing Theory. North-Holland, New York, NY, second edition, 1981.

[15] L. Dai, C.H. Chen, and J.R. Birge. Convergence properties of two-stage stochastic programming. Journal of Optimization Theory and Applications, 106(3):489–509, 2000.

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

[17] I. Gurvich, J. Luedtke, and T. Tezcan. Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach. Management Science, 56 (7):1093–1115, 2010.

[18] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. Operations Research, 29:567–588, 1981.

[19] G. Jongbloed and G. Koole. Managing uncertainty in call centers using Poisson mixtures. Applied Stochastic Models in Business and Industry, 17:307–318, 2001.

[20] O. Jouini, G. Koole, and A. Roubos. Performance indicators for call centers with impatient customers. IIE Transactions, 45(3):341–354, 2013.

[21] J.E. Kelley. The cutting-plane method for solving convex programs. Journal of the Society for Industrial and Applied Mathematics, 8(4):703–712, 1960.

[22] G. Koole. Call center mathematics. In preparation, 2005.

[23] M. Lafond. Using Erlang C to compare PSAPS. Public safety commucation, pages 30–38, January 2012.

[24] A. M. Law. Simulation Modeling & Analysis. McGraw-Hill, Boston, MA, USA, 4th edition, 2007.

[25] P. L'Ecuyer. SSJ: A Java Library for Stochastic Simulation, 2004. Software user's guide. Available at: `http://www.iro.umontreal.ca/~lecuyer`.

[26] P. L'Ecuyer and E. Buist. Simulation in Java with SSJ. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, Proceedings of the 2005 Winter Simulation Conference, pages 611–620, Piscataway, NJ, 2005. IEEE Press.

[27] P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton. An object-oriented random-number package with many long streams and substreams. Operations Research, 50 (6):1073–1075, 2002.

[28] B. G. Lewis and R. D. Herbert. Simulating the call streams to an emergency services call centre. In The 6th International Conference on Information Technology and Applications, pages 259–264. ICITA, 2009.

[29] B. G. Lewis, R. D. Herbert, P. F. Summons, and W. J. Chivers. Agent-based simulation of a multi-queue emergency services call centre to evaluate resource allocation. In MODSIM, 2007.

[30] B. Oreshkin, P. L'Ecuyer, and N. Régnard. Rate based arrival process models for modeling and simulation of call centers. manuscript, 2013.

[31] P. Reynolds. Call center metrics: Best practices in performance measurement and management to maximize quitline efficiency and quality. North American Quitline Consortium, 2010.

[32] T. Seyrafiaan. Operational Indicators – Sevice Levels & ASA. Customer Reach Newsletter June 2010, July 25th, 2010.

[33] T. Seyrafiaan. Workforce management demystified. Customer Reach Newsletter June 2011, July 26th, 2011.