Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, P. Lendermann, eds.

A LOGISTIC REGRESSION AND LINEAR PROGRAMMING APPROACH FOR MULTI-SKILL STAFFING OPTIMIZATION IN CALL CENTERS

Thuy Anh Ta

Department of Computer Science
Phenikaa University
Hanoi, VIETNAM

Tien Mai

School of Computing and Information Systems
Singapore Management University
SINGAPORE

Fabian Bastin

Department of Computer Science
and Operations Research
University of Montreal, CANADA

Pierre l'Ecuyer

Department of Computer Science
and Operations Research
University of Montreal, CANADA

ABSTRACT

We study a staffing optimization problem in multi-skill call centers. The objective is to minimize the total cost of agents under some quality of service (QoS) constraints. The key challenge lies in the fact that the QoS functions have no closed-form and need to be approximated by simulation. In this paper we propose a new way to approximate the QoS functions by logistic functions and design a new algorithm that combines logistic regression, cut generations and logistic-based local search to efficiently find good staffing solutions. We report computational results using examples up to 65 call types and 89 agent groups showing that our approach performs well in practice, in terms of solution quality and computing time.

1 INTRODUCTION

This paper concerns the management of staffing in a call center (or in general a contact center), which can be generally defined as a central office used for receiving or transmitting customers' requests by telephone, computer, or other office equipment. Agents in a contact center work with customers to resolve complaints, process orders, and provide information about an organization's products and services. They typically provide service by phone, email, text, or even through social media. Contact centers are very popular in society and can be found in many customer care services, fire alarms, telemarketing, etc. The contact center industry is continuously growing and playing an important role in society. For instance, in the United States, in 2020, the contact center industry created about 2,999,400 jobs with a median pay of about $35,830 per year, according to the Bureau of Labor Statistics in 2020 (https://www.bls.gov/ooh/office-and-administrative-support/customer-service-representatives.htm#tab-1). For more about call/contact centers, we refer the reader to Gans et al. (2003), Wallace and Whitt (2005), Avramidis and L'Ecuyer (2005), and Koole (2013).

In this paper we focus on the staffing optimization problem in multiskill contact center where the objective is to minimize the staffing cost while ensuring that some quality of service (QoS) constraints are satisfied. This is one of the essential problems in the management of a contact center. In a multiskill setting, i.e., agents have multi-skills and can serve different types of customers, the QoS functions do not have closed-forms and often require simulation to to be approximated. The literature has seen a number of studies making use of simulation and the cutting plane method to practically solve the staffing/scheduling

problems. For example, Atlason et al. (2004) propose a method that combines cut generation and linear programming to solve a scheduling problem in a single call type and single-skill call center with long-term expected service level (SL) constraints. Ceẓik and L'Ecuyer (2008) use the cutting plane method to solve large-size staffing problems in a single time period for multi-skill call centers. Ta et al. (2021) use the cutting plane method and a Benders decomposition to solve a two-stage stochastic staffing optimization problem under arrival rate uncertainty. The cutting plane method is based on the observation that the QoS functions often have an "S-shape" and the optimal solutions, in most cases, belong to the concave regions of the QoS functions. This suggests the idea of generating linear cuts to approximate the concave regions of the QoS functions. As highlighted in prior work (Atlason et al. 2008; Ceẓik and L'Ecuyer 2008; Ta et al. 2021), there are two issues associated with the cutting plane approach: (i) the cuts are based on simulation and may remove good solutions due to simulation noise, and (ii) the QoS functions are not concave everywhere and determining the region of concavity is difficult. This can lead to staffing solutions that are far from optimal. In this paper, we address these issues by proposing a new approach based on logistic regression to handle the nonlinear QoS constraints.

Our approach exploits the observation that the QoS function typically display "S-shaped" curves and could be well approximated by *logistic functions*. We develop a regression-based approach that combines simulation, logistic regression and linear programming to quickly find a staffing solution satisfying the QoS constraints. Moreover, observing that the approximations might be good only in some restricted regions, we design a local search procedure in which at each iteration, we approximate the QoS by logistic functions and only solve the approximated problem in a restricted area (a trust region) around the current solution candidate. With the logistic-based approximation, we show that the sub-optimization problems to be solved at each iteration of the local search can be linearly formulated and efficiently solved using an existing solver such as CPLEX. This local search procedure allows use to improve any feasible solutions returned by the regression-based model mentioned above, or the cutting plane method. Our logistic-based approach differs from the cutting plane method as our method does not require to identify the concave regions of the QoS functions. Moreover, since at each step of the local search, we only solve the approximate linear program in a restricted area around the current candidate, simulation noises would yield less impacts on the solution quality, as compared to the cutting plane method.

We design a new algorithm consisting of four main steps, namely, two steps to collect QoS values and learn the shapes of the QoS functions using logistic regression, one step to generate linear cuts to approximate the concave regions of the QoS functions (the cutting planes), and the final step to improve a feasible solution using the logistic-based local search procedure. Our optimization procedure is a sequence of steps of performing *simulations* to approximate the QoS and solving *mixed-integer linear programs*. We test our approach on two call center examples, a medium one with 6 call types 8 agent groups, and a large real-size one with 65 call types and 89 agent groups. The numerical results clearly show the practical efficiency of our approach in finding good staffing solutions with reasonable computational budgets, as compared to the conventional cutting plane method. Our method can be extended to solve other staffing/scheduling problems in other settings.

The rest of the paper is organized as follows. Section 2 presents the problem formulation and its sample average approximation version. Section 3 presents our logistic-based approach. Section 4 provides numerical results, and finally Section 5 concludes.

2 SIMULATION-BASED STAFFING OPTIMIZATION IN MULTI-SKILL CENTERS

We now give a formulation for the staffing optimization problem in multi-skill call centers. There are K call types indexed as $1,\dots K$, and I agent groups indexed as $1,\dots,I$, and one period. Each agent group may have several skills and can serve different call types. To evaluate the quality of service offered by the call center, we use *service level* (SL) in a long run (Atlason et al. 2008; Ceẓik and L'Ecuyer 2008; Avramidis et al. 2010). However, instead of imposing requirements for the expected SL, we are interested in chance constraints on the randomness of the SL. Specifically, we require that the SL targets are satisfied

for a target proportion of the days, as in Ta et al. (2016), Chan et al. (2016). An example of a chance constraint on the SL is, for example, that the probability that at least 95% of calls are answered within $\tau = 2$ seconds in a given time period is at least 85%. This constraint is used for the model of the Montreal 911 emergency call center presented in Ta et al. (2016).

Given a staffing vector x, let $S_k(\tau_k, x)$ be the fraction of calls of type k answered within τ_k seconds (the SL for call type k) for $k = 1, \ldots, K$, and $S_0(\tau_0, x)$ be the fraction of all calls answered within τ_0 seconds (the aggregated SL). All of these are random variables whose distributions depend on the entire staffing. We consider the chance constraints of the form: *the probabilities that the service levels are satisfied are no smaller than some given thresholds*. More precisely, the constraints can be written as $g_k(x) := \mathbb{P}[S_k(\tau_k, x) \geq s_k] \geq l_k$, where for $k = 0, \ldots, K$, s_k is the target of SL for call type k, $l_k \in [0,1]$ is the target for the probabilistic constraint for call type k, and $k = 0$ refers to the aggregation of all other types.

The objective is to minimize the operating cost of the center while satisfying a set of chance constraints on SL. 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. The staffing optimization problem can be formulated as $\min_{x \in \mathbb{N}^I} c^{\mathrm{T}} x$ subject to $g_k(x) \geq l_k$ for $k = 0, \ldots, K$, where $c = (c_1, \ldots, c_I)^{\mathrm{T}}$ is a cost vector, c_i is the cost of an agent in group i. In our context, the functions $g_k(x)$ are too hard to compute and must be estimated by simulation. Then we solve the approximated problem in which the $g_k(\cdot)$ are replaced by their estimates. This approach is known as *sample average approximation* (SAA). Suppose we perform M simulation runs to get the estimates of probabilities. We denote the empirical service level in the j-th replication by $\hat{S}^j_{k,M}(\tau_k, x)$ for each call type $k = 1, \ldots, K$, and $\hat{S}^j_{k,M}(\tau_k, x)$ for the aggregation ($k = 0$). We denote

$$\hat{g}_{k,M}(x) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{I}[\hat{S}^j_{k,M}(\tau_k, x) \geq s_k], \text{ for all } k = 0, \ldots, K,$$

where $\mathbb{I}$ is the indicator function. The SAA problem is:

$$(\mathbf{P1}) \qquad \begin{cases} \underset{x \in \mathbb{N}^I}{\text{minimize}} & c^{\mathrm{T}} x \\ \text{subject to} & \hat{g}_{k,M}(x) \geq l_k \qquad\qquad k = 0, \ldots, K \end{cases}$$

We also denote $g(\cdot) = (g_0(\cdot), \ldots, g_K(\cdot))$ and $\hat{g}_M(\cdot) = (\hat{g}_{0,M}(\cdot), \ldots, \hat{g}_{K,M}(\cdot))$. One can show that, under some conditions, by solving the SAA problem, almost surely we can retain a true optimal solution when the sample size M is large enough, and the probability of getting an exact solution approaches one exponentially fast when M grows to infinity (Ta et al. 2021).

A key issue when solving (**P1**) is that the functions $\hat{g}_{k,M}(x)$ are nonlinear and often not smooth because of simulation noise. In previous work (Ceẓik and L'Ecuyer 2008; Ta et al. 2021), the nonlinear functions are approximated by linear cuts, and a staffing solution is found by iteratively adding cuts and solving the resulting linear programs. As mentioned earlier, these cuts can be noisy because of the simulation noise, and sometimes eliminate a large area around the optimal solution. The method we propose here provides a new way to overcome this problem, by constructing more stable cuts.

3 THE LOGISTIC REGRESSION AND LINEAR PROGRAMMING APPROACH

This section describes our approach to approximate the QoS functions $\hat{g}_{k,M}(x)$ by logistic functions. We present a logistic regression-based staffing optimization model and a logistic-based local search method.

3.1 Approximating the QoS by Logistic Functions

We first motivate with examples the idea of using logistic functions to approximate the QoS. Previous studies (Chan et al. 2016; Ta et al. 2021) have shown empirically that with large enough sample size M, the QoS function $\hat{g}_{k,M}$ typically has the following properties:

(i) $\hat{g}_{k,M}(x)$ is a probability function, so $\hat{g}_{k,M}(x) \in [0,1]$ for all $x \in \mathbb{N}^I$;
(ii) $\hat{g}_{k,M}(0) = 0$, $\lim_{x\to\infty} \hat{g}_{k,M}(x) = 1$, and there exists a staffing vector $\hat{x}$ such that $\hat{g}_{k,M}(x) = 1$ for all $x \geq \hat{x}$;
(iii) If we fix the vector $x = x^*$ except for an element x_i, then $\hat{g}_{k,M}(x_1^*, \ldots, x_i, \ldots x_I^*)$ as a function of x_i is constant or has the shape of a logistic function.

In view of the above properties, it appears reasonable to approximate the QoS functions by logistic functions, of the form

$$h(x, \alpha_k) = 1/\left(1 + \exp(-(\alpha_k^1)^{\mathrm{T}} x + \alpha_k^0)\right), \quad k = 0, \ldots, K, \tag{1}$$

where $\alpha_k^1 \geq 0$ is a vector of size I and $\alpha_k^0 \geq 0$ is a scalar, for each k. We expect that $h(x, \alpha_k)$ should give a good fit to the function $\hat{g}_{k,M}(x)$, or equivalently that a linear function should fit well with $\ln(1/\hat{g}_{k,M}(x) - 1)$. We can then apply standard least-squares linear regression to estimate the vector α by fitting this function to the linear function $(-(\alpha_k^1)^{\mathrm{T}} x + \alpha_k^0)$. However, $\log(1/\hat{g}_{k,M}(x) - 1)$ is undefined when $\hat{g}_{k,M}(x)$ is 0 or 1. To get around this problem, we define

$$\nu_{k,M}(x) = \begin{cases} \hat{g}_{k,M}(x) & \text{if } 0 < \hat{g}_{k,M}(x) < 1, \\ \nu_{k,M}(x) = \nu_1 & \text{if } \hat{g}_{k,M}(x) = 0, \\ \nu_{k,M}(x) = \nu_2 & \text{if } \hat{g}_{k,M}(x) = 1, \end{cases}$$

where ν_1 and ν_2 are two constants such that ν_1 is close to 0 and ν_2 is close to 1, but not too close, to avoid numerical issues. Note that since $\hat{g}_{k,M}(x)$ is the average of M indicator functions, if $0 < \hat{g}_{k,M}(x) < 1$, then we always have $\hat{g}_{k,M}(x) \in [1/M, 1 - 1/M]$. We will use these $\nu_{k,M}(x)$ throughout the rest of the paper.

The least-squares fitting is done by solving

$$(\mathbf{P2}) \qquad \underset{\alpha_k \in \mathbb{R}^{I+1}}{\text{minimize}} \qquad \frac{1}{T} \sum_{t=1}^{T} w_k(x^t) \left(-(\alpha_k^1)^{\mathrm{T}} x^t + \alpha_k^0 - \log\left(\frac{1}{\nu_{k,M}(x^t)} - 1 \right) \right)^2$$

which yields a closed form optimal solution $\alpha_k = (\mathscr{X}^{\mathrm{T}} \mathscr{X})^{-1} \mathscr{X}^{\mathrm{T}} \tilde{g}^k$, where $\mathscr{X}$ is matrix of size $T \times I$ whose t-th row is vector $(\sqrt{w_t(x^t)}(x^t)^{\mathrm{T}}, \sqrt{w_t(x^t)})$ and $\tilde{g}^k$ is a vector of size T with t-th element $\tilde{g}_t^k = \sqrt{w_k(x^t)} \log(1/\nu_{k,M}(x^t) - 1)$.

We illustrate this with two examples, both based on a small call center model with two agent groups and two call types. Group 1 can serve only call type 1, and Group 2 can serve both call types (1 and 2). Agents in group 2 prioritize calls of type 2 over those of type 1, and arriving calls of type 1 are first routed to idle agents in group 1, if any. We also assume that, for the two call types, each caller abandons immediately with probability 2% if it has to wait and patience times are exponential with means 10 and 6 minutes. The length of the period is one hour. The parameters in the SL constraints are $\tau_1 = \tau_2 = \tau_0 = 120$ (seconds), $s_1 = s_2 = 80\%$ and $s_0 = 85\%$. We consider the SAA $\hat{g}_{k,M}(x)$ for $k = 2$, using $M = 1000$.

Example 1: Fitting $\hat{g}_{2,M}(x)$ and $\ln(1/\hat{g}_{2,M}(x) - 1)$ with different logistic and linear functions. We evaluated the function $\hat{g}_{2,M}(x)$ at a staffing (for Group 2) $x_2 \in \{4, 5, \ldots, 16\}$ and $x_1 = 15$ for Group 1. On the left side of Figure 1, we plot $\hat{g}_{2,M}$ and the functions of the form $1/(1 + \exp(-\alpha_1 x_2 + \alpha_2))$, and on the right side we plot $\ln(1/\hat{g}_{2,M}(x_2) - 1)$ and linear functions $-\alpha_1 x_2 + \alpha_2$, where $\alpha \in \Big\{(1.14, 9.9), (1.14, 10.9), (1.24, 9.9)\Big\}$. Clearly, with $\alpha = (1.14, 9.9)$, the corresponding logistic and linear functions seem to fit very well with $\hat{g}_{2,M}(x_2)$ and $\ln(1/\hat{g}_{2,M}(x_2) - 1)$, in particular with $x_2 \in [6, 14]$ and $\hat{g}_{2,M}(x_2) \in [0.03, 0.998]$. Note that the values $\alpha = (1.14, 9.9)$ are obtained by fitting the linear function with 13 values of $1/(1 + \exp(-\alpha_1 x_2 + \alpha_2))$ (evaluated at $x_2 \in \{4, 5, \ldots, 16\}$).

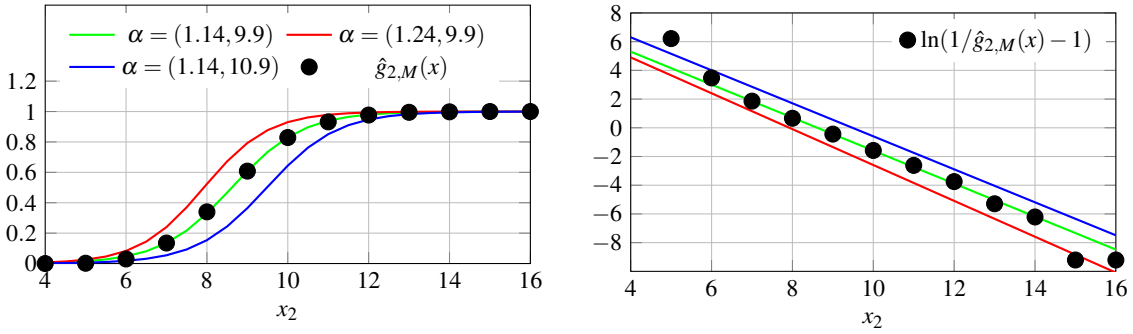

Figure 1: Fitting $\hat{g}_{2,M}(x)$ with logistic and $\ln(1/\hat{g}_{2,M}(x) - 1)$ with linear functions..

Example 2: The shapes of $\hat{g}_{1,M}(x)$ and $\ln(1/\hat{g}_{1,M}(x) - 1)$ in 3D. This example illustrates how $\hat{g}_{1,M}(x)$ and $\ln(1/\hat{g}_{1,M}(x) - 1)$ look in 3D, noting that call type 1 can be served by both agent groups. We vary both x_1, x_2, so function $\hat{g}_{1,M}(x)$ becomes $\hat{g}_{1,M}(x_1, x_2)$. We compute 23×32 values of $\hat{g}_{1,M}(x_1, x_2)$, where $x_1 \in \{1, \ldots, 23\}$ and $x_2 \in \{1, \ldots, 32\}$. We draw the 3D surface plots given by $\hat{g}_{1,M}(x_1, x_2)$ and $\ln(1/\hat{g}_{1,M}(x_1, x_2) - 1)$ in Figure 2, noting that when computing $\ln(1/\hat{g}_{1,M}(x_1, x_2) - 1)$, if $\hat{g}_{1,M}(x_1, x_2) = 0$ or 1 then we replace it by 0.0001 and 0.9999 to avoid numerical issues. We see that $\hat{g}_{1,M}(x_1, x_2)$ on the left hand side seems to have a shape of a logistic function. On the other hand, $\ln(1/\hat{g}_{1,M}(x_1, x_2) - 1)$ has approximately a linear shape in the area where the $\hat{g}_{1,M}(x_1, x_2)$ are not too close to 0 or 1, e.g., $\hat{g}_{1,M}(x_1, x_2) \in [0.001, 0.995]$.

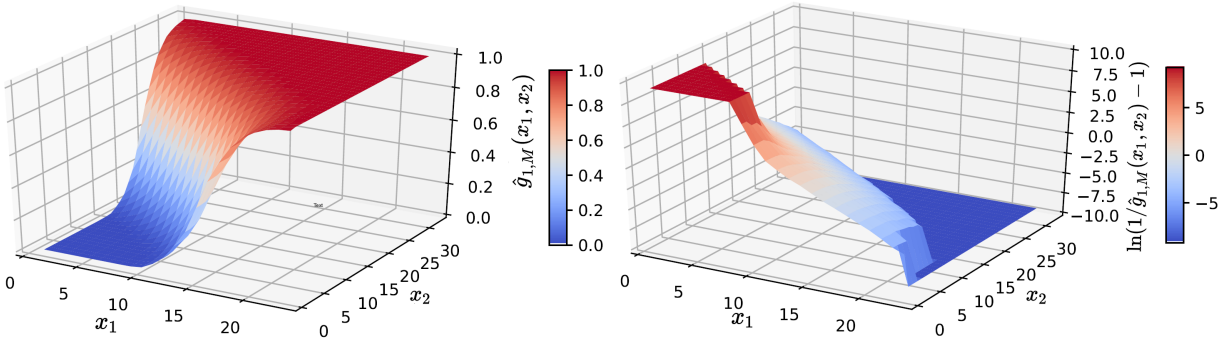

Figure 2: 3D surface plots of $\hat{g}_{1,M}(x_1, x_2)$ and $\ln(1/\hat{g}_{1,M}(x_1, x_2) - 1)$

3.2 Logistic Regression-based Optimization Model

When the logistic model $h(x, \alpha_k)$ can provide a good approximation of the QoS function $\hat{g}_{k,M}(x)$, we can replace the chance constraints by constraints on $h(x, \alpha_k)$, which can be transformed into linear ones. Let $\alpha_k^* = \{(\alpha_k^{1*}, \alpha_k^{0*})\}$ denote the set of parameters obtained after fitting function $h(x, \alpha_k)$ with $\hat{g}_{k,M}(x)$. Given these parameters, we can replace the constraints in (**P1**) by $h(x, \alpha_k^*) \geq l_k$, for all $k = 0, \ldots, K$, and these constraints can be transformed equivalently into linear ones as $-(\alpha_k^{1*})^{\mathrm{T}} x + \alpha_k^{0*} \leq \log(1/l_k - 1)$ for $k = 0, \ldots, K$. Then, (**P1**) can be equivalently formulated as the integer linear programming problem:

$$(\mathbf{P3}) \qquad \begin{cases} \underset{x \in \mathbb{N}^I}{\text{minimize}} & c^{\mathrm{T}} x \\ \text{subject to} & (\alpha_k^{1*})^{\mathrm{T}} x \geq \alpha_k^{0*} - \log(1/l_k - 1), \quad k = 0, \ldots, K. \end{cases} \tag{2}$$

Problem (**P3**) can then be solved conveniently using a commercial solver such as CPLEX.

The regression-based optimization model (**P3**) can be combined with the fitting procedure in an iterative manner to find good staffing solutions, as follows. At iteration t, we add $(x^t, \hat{g}_{k,M}(x^t))$ to the training set of the regression model and update α_k^*, $k \in \{0,\dots,K\}$. To obtain new solutions, we replace $\hat{g}_{k,M}(x^t)$ by $h(x,\alpha_k^*)$ and solve the approximate problem (**P3**) with constraints $h(x,\alpha_k^*) \geq l_k$, $k = 0,\dots,K$. When the training set has enough points, the parameter estimates α_k^*, $k \in \{0,\dots,K\}$ become stable and we can stop the iterative procedure and return the best solution found. In general, this approach does not require estimating the sub-gradients as in the conventional cutting plane method, so it is generally less expensive.

3.3 Logistic-based Trust Region Local Search

We now outline a local search procedure that allows to improve any feasible staffing solution. The algorithm incorporates the idea of the trust region method used in continuous optimization (Nocedal and Wright 2006). This is an iterative procedure in which, at each iteration, we build model functions approximating the QoS functions and define a region around the current solution within which we trust the model functions to be adequate representations of the QoS. Then, we find a next solution candidate by minimizing the optimization problem in which the QoS are replaced by the model functions, and inside the region we trust, we hope to find a new candidate solution with better objective value. The size of the region is reduced or enlarged according to the quality of the new solution found.

The difference between our approach and other conventional trust region algorithms in the literature is that we build the model functions based on the idea of approximating the QoS functions by *logistic functions*. The idea is to approximate the $\log(1/\nu_{k,M}(x)-1)$ by linear functions of x, and build model functions based on the estimated gradients of $\log(1/\nu_{k,M}(x)-1)$.

More precisely, let us define $v_k(x) = \log(1/\nu_{k,M}(x)-1)$, $k = 0,\dots,K$. Given a point $\bar{x}$, let $\nabla v_k(\bar{x})$ denote a (tentative) estimation of the subgradient $v_k(.)$ at $\bar{x}$. The vector $v_k(\bar{x})$ has no closed-form and need to be approximated by simulation. Similar to the cutting plane approach, the i-th element of $\nabla v_k(\bar{x})$ can be approximated by finite difference as

$$\nabla v_{ki}(\bar{x}) = \frac{v_k(\bar{x}+de_i)-v_k(\bar{x})}{d} = \frac{1}{d}\log\left(\frac{(1-\nu_{kM}(\bar{x}+de_i))\nu_{kM}(\bar{x})}{(1-\nu_{kM}(\bar{x}))\nu_{kM}(\bar{x}+de_i)}\right),\ k = 0,\dots,K, \tag{3}$$

where d is a step size. We normally choose $d = 1$, but we may need to increase d, e.g., $d = 2,3$, to avoid simulation noise when the number of samples M is not large enough.

Given the vector $\nabla v_k(\bar{x})$, we can approximate $v_k(x)$ by a linear model $m_k(x)$ such that $m_k(\bar{x}) = v_k(\bar{x})$ and $\nabla m_k(\bar{x}) = \nabla v_k(\bar{x})$, $\forall k = 1,\dots,K$, which leads to

$$v_k(x) \approx m_k(x) = v_k(\bar{x}) + \nabla v_k(\bar{x})(x-\bar{x}). \tag{4}$$

Quadratic model functions are more commonly used in the trust region literature. However, in our empirical experiments, linear functions turned out to be a better choice (in our context) to approximate $\ln(1/\hat{g}_{k,M}(.)-1)$. Moreover, as shown in the following, linear model functions yield linear sub-problems, which are practically more convenient to deal with. Our trust-region local search algorithm works as follows. We start with a feasible solution, i.e., a solution that satisfies the QoS constraints. At each iteration t with solution x^t we define a model function $m_k(x)$ as in (4) and a region where we have confidence that the model approximates the objective function well enough. We then solve the optimization model with constraints on $m_k(x)$ to find a new solution $\bar{x}^t$. If $\bar{x}^t$ satisfies the chance constraints and gives lower cost, i.e., $c^{\mathrm{T}}\bar{x}^t < c^{\mathrm{T}}x^t$, then we update $x^{t+1} = \bar{x}^t$ and enlarge the trust-region radius. Otherwise, we keep the current solution, i.e., $x^{t+1} = x^t$, and reduce the trust region. We stop the algorithm when none of the operations result in a strict decrease in the agent cost.

To obtain a new solution using the model function $m_k(x)$, we seek a solution to the following sub-problem

$$(\mathbf{P4}) \quad \begin{cases} \underset{x\in\mathbb{N}^I}{\text{minimize}} & c^{\mathrm{T}}x \\ \text{subject to} & m_k(x) = v_k(x^t) + \nabla v_k(x^t)(x-x^t) \leq \log(1/l_k - 1) \\ & ||x-x^t|| \leq \Delta_t \end{cases} \tag{5}$$

where Δ_t is the trust-region radius at iteration t, and $||x-x^t||$ is a norm of vector $x-x^t$. We choose the L_1-norm for the trust-region constraints (5) so that these constraints can be linearized conveniently using auxiliary variable $z \in \mathbb{R}^I$ as (i) $x_i - x_i^t \leq z_i$, (ii) $x_i^t - x_i \leq z_i$ and (iii) $\sum_i z_i \leq \Delta_t$. Constraints (5) can also be linearized with L_∞-norm, but we use L_1-norm to have smaller trust regions. Moreover, since we only seek integer solutions, we should have $\Delta_t \geq 1$.

The above mixed-integer linear program (MILP) can be handled conveniently using a MILP solver such as CPLEX. During the local search procedure, we iteratively solve (**P4**) to get new solutions. Note that x^t is feasible to (**P4**) for any $\Delta_t \geq 0$, so if $\bar{x}^t$ is an optimal solution to (**P4**), then we always have $c^{\mathrm{T}}\bar{x}^t \leq c^{\mathrm{T}}x^t$. Moreover, if we find a solution $\bar{x}^t$ by solving (**P4**) and $\bar{x}^t$ does not satisfy the chance constraints, then we need to reduce the trust region radius Δ_t to improve the accuracy of the model $m_k(x)$. In the case that $\Delta_t \leq 1$ but we still cannot find a solution that is feasible to the QoS constraints and giving a better agent cost, then we can stop the local search procedure.

3.4 Algorithm

We summarize our approach in Algorithm 1. The algorithm has four main steps. In Step 1, we start with an initial staffing solution given by the fluid model. Then, we select call types for which the QoS constraints are not satisfied and we add more agents to the groups that serve these call types. This process allows to improve the QoS values until all the QoS constraints are satisfied. After Step 1, we can get one feasible solutions and a set S containing several staffing vectors and their corresponding QoS values.

In Step 2, we use this set to estimate the parameters α of the logistic functions, and iteratively solve the regression-based optimization model (**P3**) to (hopefully) get a good staffing solution. In Step 3, we use the cutting plane method (Atlason et al. 2004; Atlason et al. 2008) to create piece-wise linear and concave functions that outer-approximate the concave parts of the QoS functions. The cutting plane procedure consists of two main steps, namely, a simulation step to compute subgradient vectors given, and a step of adding a linear cut to for each QoS value that does not satisfy the QoS constraints. We stop the cutting procedure when all the constraints are satisfied, or we find a staffing vector giving a higher cost than the solution found from Step 2. The latter occurs when the linear cuts generated are not good and eliminate good staffing solutions. In this situation, the cutting plane method cannot return a better solution than the regression-based approach.

After Steps 1, 2 and 3, we obtain a solution that is feasible to the QoS constraints. The final step allows to further improve that solution by searching around its neighbourhood. After getting an estimation of the gradient of $v_k(x^t)$ (Step 4.1), we replace the QoS functions by the approximate model $m_k(\cdot)$ and solve the corresponding sub-problem to find a new candidate solution. In case the solution is not feasible, which means that the approximate models $m_k(.)$ do not provide good approximations to the QoS within the region identified by Δ_t, we need to reduce Δ_t to improve the accuracy of $m_k(.)$. Moreover, if we find a solution that is identical to the current one x^t, then we can stop the local search, as one can show that we cannot find a better solution by just reducing the trust region. On the contrary, if we obtain a solution that is feasible to the QoS constraints and giving a better cost as compared to the current one x^t, we move to that better solution and continue the local search, and enlarge the trust region. In general, the algorithm stops when we are in the situation that the local search cannot further improve the current solution.

Algorithm 1:

 # Step 1. Collect QoS values and find a staffing solution by the regression-based optimization

Select a step size $d, s \in \mathbb{N}$ and an initial solution x. Set $S = \emptyset$

repeat

1.1. Select $\bar{k} = \text{argmin}_k \ \hat{g}_{k,M}(x)$, and randomly and uniformly select $i \in \mathscr{G}_{\bar{k}}$

1.2. Set $x_i \leftarrow x_i + s$ and compute $\hat{g}_{k,M}(x)$, $k = 0, \ldots, K$, via simulation

1.3. Update the training set $S = S \cup \{(x, \hat{g}_M(x))\}$

until $\hat{g}_{k,M}(x) \geq l_k$, *for all* $k = 0, \ldots, K$;

 # Step 2. Solve the logistic regression-based optimization model (P3) to find a staffing solution

repeat

2.1. Solve (**P2**) using the training set S to obtain parameters α

2.2. Solve the regression-based optimization model (**P3**) to get a new solution $\bar{x}$

2.3. Simulate to obtain $\hat{g}_{k,M}(\bar{x})$, $k = 0, \ldots, K$, and update the set $S = S \cup \{(\bar{x}, \hat{g}_M(\bar{x}))\}$

until *We find some feasible solutions*;

Denote by x^* the best feasible solution found so far

 # Step 3. Cut generation

3.1 Given any solution candidate, using simulation to compute the QoS

3.2 Add sub-gradient cuts for any call type k such that $\hat{g}_{k,M} < l_k$ to a master problem, and solve the master problem to find new candidate staffing solutions

3.3. Stop the algorithm when a feasible solution to (**P1**) is found

 # Step 4. Local search

Set $t = 0$, denote by x^0 the best feasible solution found so far, choose an initial radius Δ_t. Select $0 < \delta_1 < 1 < \delta_2$.

repeat

4.1. Compute $\nabla v_k(x^t)$ by (3)

repeat

4.2. Solve the trust-region sub-problem (**P4**) and obtain $\bar{x}$, compute $\hat{g}_{k,M}(\bar{x})$, $k = 0, \ldots, K$ via simulation

4.3. **if** $\exists k$ *such that* $\hat{g}_{k,M}(\bar{x}) < l_k$ **then**

$\Delta_t \leftarrow \delta_1 \times \Delta_t$ *# reduce the trust region*

else

If $c^{\mathrm{T}}\bar{x} < c^{\mathrm{T}}x^t$, then $\Delta_t \leftarrow \delta_2 \times \Delta_t$ *# enlarge the trust region*

Until $\Delta_t < 1$ *or* $\bar{x} = x^t$ *or* ($c^T\bar{x} < c^T x^t$ *and* $\hat{g}_{k,M}(\bar{x}) \geq l_k, \ \forall k$);

4.4. If $c^{\mathrm{T}}\bar{x} < c^{\mathrm{T}}x^t$ and $\hat{g}_{k,M}(\bar{x}) \geq l_k, \ \forall k$, then set $x^{t+1} = x^t$, $t \leftarrow t + 1$.

Until $\Delta_t < 1$ *or* $\bar{x} = x^t$;

Return x^t.

4 NUMERICAL EXPERIMENTS

We present experimental results using two multi-skill call center examples, namely, a medium-sized example with 6 call types and 8 agent groups, and a large-sized call center of 65 call types and 89 agent groups. The latter is inspired by a large real-life call center operated by Bell Canada. These examples were used in previous staffing optimization studies (Ceẓik and L'Ecuyer 2008; Ta et al. 2021).

4.1 Experimental Settings

We test our algorithm with targets $l_0 = 85\%$ and $l_k = 80\%$. The agents costs are defined based on the number of skills in the agent's skill set as $c_i = 1 + 0.1(|\mathscr{S}_i| - 1)$ for each group i, where $|\mathscr{S}_i|$ is the cardinality of $\mathscr{S}_i$. We assume that calls arrive according to stationary Poisson processes. For each call center example, we define 10 different instances by varying the arrival rates. In practice, the arrival rates depend on many factors such as the day of the week, time of the day, level of business, holidays and special events, are also random (Channouf et al. 2007; Ibrahim et al. 2016; Oreshkin et al. 2016).

We solve each instance by Algorithm 1, denoted by RCLS, an abbreviation for *Regression*, *Cutting Plane* and *Local Search*). We also try the cutting plane method (CP) as in Ceẓik and L'Ecuyer (2008), Ta et al. (2021), and the regression-based optimization approach defined by Steps 1–2 of Algorithm 1, denoted by RO. We always take sample size $M = 1000$. Moreover, the same set of realizations used to approximate the QoS is reused during the optimization process. The solutions obtained by different approaches are then evaluated via an out-of-sample study. That is, for each solution x, we compute the corresponding QoS values with sample size $M' = 2000$ that are independent of those used to obtain the solutions. Then, we report the average number of QoS values that violate the requirements, i.e., for which $\hat{g}_{k,M'}(x) \leq l_k - \kappa$, $k = 0, \ldots, K$. Here, we use two values of κ, namely, $\kappa = 0$ and $\kappa = 0.005$. The former refers to the exact QoS constraints, while for the latter we relax a bit the QoS requirements.

The CP method is implemented similarly as in Ta et al. (2021). For the RO method, we use the fluid scheduling model as in Ceẓik and L'Ecuyer (2008) to obtain an initial staffing vector to start collecting QoS values. The step size is chosen as $s = 1$ and we run Step 1 simultaneously on 8 physical CPUs to get as many points as possible. For estimating the α parameters, we select the weight vector as $w_k(x^t) = 4$ if $|\hat{g}_{k,M}(x^t) - l_k| < 0.05$ and $w_k(x^t) = 1$ otherwise. For Step 4 of Algorithm 1, we select $\Delta_0 = 8$ as an initial trust region radius. The parameters to enlarge and reduce the trust region are chosen as $\delta_1 = 0.7$ and $\delta_1 = 1.3$. These parameters are chosen manually to achieve good performance for Algorithm 1. For the definition of $v_{k,M}(\cdot)$, note that with sample size $M = 1000$, if $\hat{g}_{k,M}(x^t) > 0$ and $\hat{g}_{k,M}(x^t) < 1$, then $\hat{g}_{k,M}(x^t) \in [0.001, 0.999]$, so we choose v_1, v_2 such that $v_1 < 0.001$ and $v_2 > 0.999$. In this experiment we choose $v_1 = 0.0001$ and $v_2 = 0.9999$. The sub-gradient $\nabla v_k(x^t)$ are estimated with step size $d = 1$. The "*repeat-until*" in Step 2 stops when we find 5 feasible solutions and we just return the best one found. In general, since there are quite a lot of points generated after Step 1, Step 2 finishes after just a few iterations. We use these settings for both RO and RCLS methods.

We use the solver *cplexmilp* from CPLEX to solve mixed-integer linear programming (MIP) models under default settings. For the medium instances, we let CPLEX run to optimality and for the large instances, the relative optimality gap was set to 0.05%. To solve the linear least-squares problem (**P2**), we use *lsqlin* from MATLAB 2015a. The experiments were conducted on a machine running Debian 8 with Intel(R) Xeon(R) CPU E5620 (2.40GHz). The simulations were performed using the *ContactCenters* simulation library (Buist and L'Ecuyer 2005), developed with the SSJ simulation package (L'Ecuyer and Buist 2005).

4.2 Medium-sized Call Center

We first report numerical results for the medium call center with 6 call types and 8 agent groups in Table 1 where the best costs obtained by the three approaches are indicated in bold. We also emphasize in red the costs that are remarkably higher than the others for each instance.

		Agent cost			CPU time (hour)			Out-of-sample # violated QoSs					
								$\kappa = 0$			$\kappa = 0.005$		
Targets	Instances	RO	CP	RCLS	RO	CP	RCLS	RO	CP	RCLS	RO	CP	RCLS
(80%,85%)	1	197.0	198.2	**195.8**	0.24	0.23	0.9	0	0	0	0	0	0
	2	189.5	189.5	**189.2**	0.14	0.12	0.46	0.1	0	0	0	0	0
	3	189.4	188.9	**188.5**	0.13	0.08	0.42	0	0	0	0	0	0
	4	196.1	264.3	**195.1**	0.19	0.16	0.74	0	0	1	0	0	0
	5	182.5	182.4	**182.0**	0.15	0.23	0.59	2	0	0	0	0	0
	6	189.6	235.5	**189.4**	0.15	0.08	0.32	2	0	0	0	0	0
	7	175.0	176.7	**174.1**	0.12	0.22	0.58	0	0	0	0	0	0
	8	159.9	160.4	**158.7**	0.04	0.23	0.67	0	0	0	0	0	0
	9	192.4	212.5	**192.2**	0.11	0.3	0.32	2	0	1	1	0	0
	10	209.9	287.5	**209.2**	0.22	0.17	0.82	2	0	0	0	0	0
	Average				**0.15**	0.18	0.58	0.81	0	0.2	0.1	0	0

Table 1: Agent costs, CPU times and out-of-sample results for the medium call center examples

In terms of agent cost we generally see that the RO returns better costs than the CP in 7/10 instances. Moreover, in 4/10 instances, the CP gives very high costs as compared to the two other approaches. This clearly indicates the instability of the CP and can be explained by the issue that the fluid constraints are not be able to eliminate all the non-concave points, leading to poor sub-gradient cuts. The RCLS approach basically gives the best costs for all the instances. More precisely, RCLS improves the agent costs by 8.69% over CP and by 0.38% over RO, on average. Moreover, RO improves the agents costs by 8.33% over CP.

In terms of CPU time, RO is the fastest method as expected, and RCLS requires remarkably higher CPU times compared to the two other approaches. Even so, the RCLS requires only half an hour on average to solve one instance, which is viable in practice.

In the out-of-sample study with $\kappa = 0$, we observe some violated QoS constraints. It is interesting to see that the average numbers of violated QoS given by the CP are less than the other approaches. However, if we relax a bit the QoS constraints, i.e., $\hat{g}_{k,M}(x) \geq l_k - 0.005$, then there is no violated QoS for all the solutions returned by the three approaches, except for one RO solution. In general, we can see that RCLS seems to return solutions under which the QoS values are very close to the targets. Moreover, a sample size of $M = 1000$ seems large enough to ensure that the QoS values only vary in small intervals in different sets of realizations.

4.3 Large-sized Call Center

We now consider a larger call center model with $K = 89$ call types and $I = 65$ agent groups, inspired by a real-life call center previously operated by Bell Canada, and available at http://www.iro.umontreal.ca/~lecuyer/myftp/ld-example2/. This example is also used in Ceşik and L'Ecuyer (2008) and we refer the reader to this paper for more details. Table 2 reports numerical results for 10 instances. We also indicate in bold the best costs obtained. We can see that CP is more stable and always gives better costs than RO. However, RCLS always returns the best costs among the three approaches, for all instances. We can also see that RCLS improves the agent costs by 1.55% over CP and by 4.66% over RO, on average. Moreover, CP gives about 3.16% better agent costs than RO on average.

In terms of CPU time, RO is obviously much faster than the other approaches. This is because RO does not need to compute the sub-gradients q_k and ∇v_k, which requires 8 and 89 simulations for the medium and large examples, respectively. The CPU times required by RCLS are about 43% higher than for CP. Even so, RCLS needs less than four hours to solve one instance, which is viable in practice.

The out-of-sample results are not surprising for these large instances. Similarly to the medium call center, we observe some violated QoS constraints with $\kappa = 0$, but their number is very small, considering

		Agent cost			CPU time (hour)			Out-of-sample # violated QoSs					
								$\kappa = 0$			$\kappa = 0.005$		
Targets	Instances	RO	CP	RCLS	RO	CP	RCLS	RO	CP	RCLS	RO	CP	RCLS
(80%,85%)	1	873.1	836.7	**832.1**	0.44	1.69	1.89	0	0	0	0	0	0
	2	842.8	819.6	**801.6**	0.42	1.82	2.50	1	0	0	0	0	0
	3	861.8	834.7	**819.0**	0.46	2.71	2.31	0.1	0	0.7	0	0	0
	4	821.6	790.7	**779.9**	0.41	1.58	2.55	1	0	1.5	0	0	0
	5	869.6	850.3	**841.0**	0.39	1.93	2.43	0	0	0	0	0	0
	6	831.0	806.2	**798.3**	0.40	2.08	3.68	1	0	0	0	0	0
	7	896.2	870.8	**847.9**	0.38	2.49	2.17	0	0	0	0	0	0
	8	838.7	810.5	**797.2**	0.37	1.59	2.44	0	0	0	0	0	0
	9	831.8	796.2	**793.1**	0.35	1.69	3.19	0.9	0.1	0.1	0	0	0
	10	868.7	850.3	**827.0**	0.39	2.62	1.78	0	0.5	1	0	0	0
Average					**0.40**	2.02	2.49	0.4	**0.06**	0.33	0	0	0

Table 2: Agent costs, CPU times and out-of-sample results for the large call center examples

that there are about 66 QoS constraints to satisfy. When we relax a bit the requirements with $\kappa = 0.005$, there is no violated QoS constraint. We also observe that the average number of violated QoS values with RCLS is higher than for CP, indicating that the QoS values given by the RCLS solutions are tighter with the targets, as compared to those from CP.

5 CONCLUSION

We have introduced a new approach to solve the chance-constrained staffing optimization in a multi-skill call center. Our methodology is based on the observation that the QoS functions generally display "S-shapes", so they can be approximated by appropriate logistic functions. We have designed an algorithm that combines a regression-based step to collect QoS values and approximate QoS functions by logistic ones, a step that generates linear cuts to approximate the QoS constraints, and a local search procedure to further improve feasible solutions. We tested this approach on two call center examples, one with medium and one with large numbers of agent groups and call types. The numerical results show the efficiency of our approach in finding good staffing optimization in reasonable computing time. Our methodology is general, in the sense that it can be applied in other settings, e.g., the staffing problem with SL constraints considered in Ceẓik and L'Ecuyer (2008) or the scheduling problem in Avramidis et al. (2010). It might be also promising for larger-scale problems, e.g., a staffing or scheduling optimization problem under uncertainty (Ta et al. 2021).

ACKNOWLEDGMENTS

This work has been supported by a Canada Research Chair, an Inria International Chair, and a Hydro-Québec research grant to P. L'Ecuyer, and by NSERC Discovery Grants to F. Bastin and P. L'Ecuyer.

REFERENCES

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

Atlason, J., M. A. Epelman, and S. G. Henderson. 2008. "Optimizing Call Center Staffing using Simulation and Analytic Center Cutting Plane Methods". *Management Science* 54(2):295–309.

Avramidis, A. N., W. Chan, M. Gendreau, P. L'Ecuyer, and O. Pisacane. 2010. "Optimizing Daily Agent Scheduling in a Multiskill Call Centers". *European Journal of Operational Research* 200(3):822–832.

Avramidis, A. N., and P. L'Ecuyer. 2005. "Modeling and Simulation of Call Centers". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 144–152: IEEE Press.

Buist, E., and P. L'Ecuyer. 2005. "A Java Library for Simulating Contact Centers". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 556–565: IEEE Press.

Ceşik, M. T., and P. L'Ecuyer. 2008. "Staffing Multiskill Call Centers via Linear Programming and Simulation". *Management Science* 54(2):310–323.

Chan, W., T. A. Ta, P. L'Ecuyer, and F. Bastin. 2016. "Two-stage Chance-constrained Staffing with Agent Recourse for Multi-skill Call Centers". In *Proceedings of the 2016 Winter Simulation Conference*, 3189–3200. Piscataway, NJ, USA: IEEE Press.

Channouf, N., P. L'Ecuyer, A. Ingolfsson, and A. N. Avramidis. 2007. "The Application of Forecasting Techniques to Modeling Emergency Medical System Calls in Calgary, Alberta". *Health Care Management Science* 10(1):25–45.

Gans, N., G. Koole, and A. Mandelbaum. 2003. "Telephone Call Centers: Tutorial, Review, and Research Prospects". *Manufacturing and Service Operations Management* 5:79–141.

Ibrahim, R., H. Ye, P. L'Ecuyer, and H. Shen. 2016. "Modeling and Forecasting Call Center Arrivals: A Literature Study and a Case Study". *International Journal of Forecasting* 32(3):865–874.

Koole, G. 2013. *Call Center Optimization*. MG books, Amsterdam.

L'Ecuyer, P., and E. Buist. 2005. "Simulation in Java with SSJ". In *Proceedings of the 2005 Winter Simulation Conference*, 611–620. Piscataway, NJ: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nocedal, J., and S. Wright. 2006. *Numerical Optimization*. second ed. Springer Science & Business Media.

Oreshkin, B., N. Régnard, and P. L'Ecuyer. 2016. "Rate-Based Daily Arrival Process Models with Application to Call Centers". *Operations Research* 64(2):510–527.

Ta, T. A., W. Chan, F. Bastin, and P. L'Ecuyer. 2021. "A simulation-based decomposition approach for two-stage staffing optimization in call centers under arrival rate uncertainty". *European Journal of Operational Research* 293(3):966–979.

Ta, T. A., P. L'Ecuyer, and F. Bastin. 2016. "Staffing optimization with chance constraints for emergency call centers". In *MOSIM 2016–11th International Conference on Modeling, Optimization and Simulation*. See http://www.iro.umontreal.ca/~lecuyer/myftp/papers/mosim16emergency.pdf.

Ta, T. A., T. Mai, F. Bastin, and P. L'Ecuyer. 2021. "On a multistage discrete stochastic optimization problem with stochastic constraints and nested Sampling". *Mathematical Programming* 190(1–2):1–37.

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

AUTHOR BIOGRAPHIES

THUY ANH TA is an Assistant Professor at the Faculty of Computer Science, Phenikaa University, Vietnam. Her research interests include simulation-based optimization, nonlinear optimization, stochastic programming, with applications in workforce management and location planning. Her email address is anh.tathuy@phenikaa-uni.edu.vn.

TIEN MAI is an Assistant Professor at School of Computing and Information Systems, Singapore Management University. His research interests include discrete choice modeling, data-driven optimization, and imitation learning, with applications in transportation modeling, revenue/workforce management, and security games. More information can be found on his web page https://sites.google.com/view/tien-mai/home. His email address is atmai@smu.edu.sg.

FABIAN BASTIN is a Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He is a member of the CIRRELT research center and the Institute for Data Valorization, and co-founder of the Fin-ML initiative. His research interests include stochastic and nonlinear optimization, data science, discrete choice modeling, and applications in finance, transportation, and energy. More information can be found on his web page http://www.iro.umontreal.ca/~bastin. His email address is bastin@iro.umontreal.ca.

PIERRE L'ECUYER is a Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He is a member of the CIRRELT and GERAD research centers. His research interests cover all areas of stochastic simulation. He has published over 300 scientific articles, and has developed various simulation software libraries. He has been a referee for 168 different scientific journals. More information can be found on his web page: http://www.iro.umontreal.ca/~lecuyer. Email: lecuyer@iro.umontreal.ca.