Robust and data-driven approaches to call centers

Dimitris Bertsimas [*]   Xuan Vinh Doan [†]

November 2008

Abstract

We propose both robust and data-driven approaches to a fluid model of call centers that incorporates random arrival rates with abandonment to determine staff levels and dynamic routing policies. Resulting models are tested with real data obtained from the call center of a US bank. Computational results show that the robust fluid model is significantly more tractable as compared to the data-driven one and producing better solutions to call centers in most experiments.

1 Introduction

Telephone call centers have become an important means for many organizations to communicate effectively with their customers. From a management perspective, the two main problems of these call centers are *staffing* and *call routing*. The staffing problem addresses how to schedule staff (agents) in different working shifts. The call routing problem addresses how incoming calls need to be routed to appropriate agents. In general, these two problems are very difficult to solve due to the high complexity of modern call centers. Gans et al. (2003) discuss in detail these issues of modeling and analyzing call centers. Most of the studies of call centers focus on a single pool of identical agents and the *square-root safety staffing rule* is generally recommended (see Gans et al. (2003), Section 4.1). For call centers with multiple customer classes and multiple agent pools, the staffing problem is more difficult since the routing problem has to be solved at the same time. Gurvich and Whitt (2006) propose a *fixed-queue-ratio routing scheme* for call centers with many agent pools. In an actual call center environment, arrival

[*] Boeing Professor of Operations Research, Sloan School of Management, co-director of the Operations Research Center, Massachusetts Institute of Technology, E40-147, Cambridge, MA 02139-4307, dbertsim@mit.edu. This research is partially supported by NSF grant 0556106.

[†] Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139-4307, vanxuan@mit.edu.

rates are random and temporally varying as opposed to the usual assumption of constant or known arrival rates (see Brown et al. (2005) and reference therein). Customer abandonment also needs to be taken into account. Harrison and Zeevi (2005) propose a staffing method for multi-class/multi-pool call centers with uncertain arrival rates with a known probabilistic structure of arrival rates and customer abandonment. Bassamboo and Zeevi (2007) take a data-driven approach using historical call arrival data to approximate the distribution of the arrival rate process for the same call center model.

Contributions and Paper Outline

In this paper, we develop a fluid model to solve both the staffing and routing problem for large multi-class/multi-pool call centers with random arrival rates and customer abandonment. Given historical data, we propose a) a data-driven and b) a robust optimization approach for this call center problem. Specifically, our contributions and structure of the paper are as follows:

(1) In Section 2, we propose a discrete fluid model for call center systems and take a data-driven approach to determine staff levels and construct appropriate dynamic routing policies. This model addresses the randomness of arrival rates and customer abandonment. We show that the resulting model is a linear optimization problem.

(2) In Section 3, we apply the robust optimization approach to the call center problem. We introduce a simple uncertainty set for arrival rates based on some structural properties of optimal queueing and routing solutions.

(3) In Section 4, we compare the performance of solutions obtained from the two approaches, data-driven and robust optimization. We report computational results from simulations for some customer-agent network designs with call center data obtained from SEESTAT software (see Trofimov et al. (2006)). These results show that the proposed robust fluid model is significantly more tractable as compared to the data-driven model and producing better solutions in most experiments.

2 A Discrete Fluid Model for Call Centers

2.1 Fluid Model Formulation

We consider general call centers with multiple customer classes and multiple agent pools. Let $\mathcal{I}$ be the set of customer classes, $|\mathcal{I}| = m$, and $\mathcal{J}$ be the set of agent pools, $|\mathcal{J}| = n$. Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be the customer-agent connectivity matrix: $a_{ij} = 1$ if agents in pool j can serve customers from class i; otherwise, $a_{ij} = 0$. We consider the discretized planning interval $[0, T]$, which is divided into T *unit* periods indexed from 1 to T. At each time t, $t = 0, \dots, T$, let $q_i(t) \geq 0$ be the number of class-i customers waiting in queue, $i \in \mathcal{I}$, and $s_{ij}(t) \geq 0$ be the number of class-i customers being served in pool j, $j \in \mathcal{J}$. In each period t, $t = 1, \dots, T$, we observe $\tilde{\lambda}_i(t)$ class-i customers arrive, $i \in \mathcal{I}$. There are $\tilde{a}_i(t)$ class-i customers who abandon the queue, $0 \leq \tilde{a}_i(t) \leq q_i(t-1)$, while $\tilde{l}_{ij}(t)$, $0 \leq \tilde{l}_{ij}(t) \leq s_{ij}(t-1)$, is the number of class-i customers leaving the system after being served in pool j, $j \in \mathcal{J}$, in period t. Under the fluid approximation scheme, we introduce the abandonment rate $\theta_i < 1$ and the service rate $\mu_{ij} < 1$ such that $\tilde{a}_i(t) = \theta_i q_i(t-1)$ and $\tilde{l}_{ij}(t) = \mu_{ij} s_{ij}(t-1)$ for all $t = 1, \dots, T$. We need to allocate $u_{ij}(t) \geq 0$ class-i customers who are in the queue to each agent pool j, $j \in \mathcal{J}$.

The system dynamics is formulated as follows:

$$\begin{cases} q_i(t) = (1-\theta_i) q_i(t-1) - \sum_{j \in \mathcal{J}} u_{ij}(t) + \tilde{\lambda}_i(t) \\ s_{ij}(t) = (1-\mu_{ij}) s_{ij}(t-1) + u_{ij}(t) \end{cases}. \tag{1}$$

for all $t = 1, \dots, T$, $i \in \mathcal{I}$, $j \in \mathcal{J}$.

In this call center problem, we need to decide the agent pool capacity $b_j(s)$ for all $j \in \mathcal{J}$ and $s = 1, \dots, S$, assuming there are S uniform shifts in the planning interval. Given the uncertainty in arrival rates, we also want to predetermine the portion $d_{ij}(t) \geq 0$ of each agent pool $j \in \mathcal{J}$ reserved for class-i customers, $i \in \mathcal{I}$, in each period t, $t = 1, \dots, T$. The dynamic routing policy will be implemented to maintain this customer-agent allocation throughout the planning interval. We then have $\sum_{i \in I} d_{ij}(t) = b_j(\lceil tS/T \rceil)$ and the capacity constraints become $s_{ij}(t) \leq d_{ij}(t)$ for all $t = 1, \dots, T$, $i \in \mathcal{I}$, $j \in \mathcal{J}$.

We would like to minimize the staffing cost, waiting and abandonment penalty. The cost function can be written as follows:

$$\sum_{s=1}^{S} \sum_{j \in J} c_j b_j(s) + \sum_{t=1}^{T} \sum_{i \in I} (k_i^q + k_i^a \theta_i) q_i(t), \tag{2}$$

where c_j, k_i^q, and k_i^a are the appropriate cost coefficients for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$.

Setting $q_i(0) = 0$ and $s_{ij}(0) = 0$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$ as initial conditions, we obtain the final discrete fluid formulation for call centers as follows:

$$\begin{array}{rll}
\min & \sum_{s=1}^{S}\sum_{j\in J} c_j b_j(s) + \sum_{t=1}^{T}\sum_{i\in\mathcal{I}}(k_i^q + k_i^a\theta_i)q_i(t) & \\
\text{s.t.} & q_i(t) = (1-\theta_i)q_i(t-1) - \sum_{j\in\mathcal{J}} u_{ij}(t) + \tilde{\lambda}_i(t), & t = 1,\ldots,T, \\
& q_i(t), u_{ij}(t) \geq 0, & t = 1,\ldots,T, \\
& s_{ij}(t) = (1-\mu_{ij})s_{ij}(t-1) + u_{ij}(t), & t = 1,\ldots,T, \\
& 0 \leq s_{ij}(t) \leq d_{ij}(t), & t = 1,\ldots,T, \\
& q_i(0) = 0, s_{ij}(0) = 0, & i \in \mathcal{I},\ j \in \mathcal{J}, \\
& \sum_{i\in I} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1,\ldots,T, \\
& d_{ij}(t) \geq 0, & t = 1,\ldots,T, \\
& b_j \geq 0, & j \in \mathcal{J}.
\end{array} \tag{3}$$

We introduce some additional definitions which will be used later in the paper. Let $\mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ be the total waiting and abandonment penalty, which is the optimal value of the following optimization problem:

$$\begin{array}{rll}
\min_{\boldsymbol{Q},\boldsymbol{S},\boldsymbol{U}} & \sum_{t=1}^{T}\sum_{i\in\mathcal{I}}(k_i^q + k_i^a\theta_i)q_i(t) & \\
\text{s.t.} & q_i(t) = (1-\theta_i)q_i(t-1) - \sum_{j\in\mathcal{J}} u_{ij}(t) + \tilde{\lambda}_i(t), & t = 1,\ldots,T, \\
& q_i(t), u_{ij}(t) \geq 0, & t = 1,\ldots,T, \\
& s_{ij}(t) = (1-\mu_{ij})s_{ij}(t-1) + u_{ij}(t), & t = 1,\ldots,T, \\
& 0 \leq s_{ij}(t) \leq d_{ij}(t), & t = 1,\ldots,T, \\
& q_i(0) = 0, s_{ij}(0) = 0, & i \in \mathcal{I},\ j \in \mathcal{J}.
\end{array} \tag{4}$$

where $\boldsymbol{D} \in \mathbb{R}^{m\times n\times T}$, $\tilde{\Lambda} \in \mathbb{R}^{m\times T}$, $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{\theta} \in \mathbb{R}^m$. The fluid model is then rewritten as follows:

$$\begin{array}{rll}
\min_{\boldsymbol{b},\boldsymbol{D}} & \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) & \\
\text{s.t.} & \sum_{i\in I} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1,\ldots,T, \\
& d_{ij}(t) \geq 0, & t = 1,\ldots,T, \\
& b_j(s) \geq 0, & s = 1,\ldots,S,
\end{array} \tag{5}$$

where $\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) = \sum_{s=1}^{S}\sum_{j\in\mathcal{J}} c_j b_j(s) + \mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$. We also notice that the penalty $\mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ is separable with respect to customer classes, $\mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) = \sum_{i\in\mathcal{I}} \mathcal{Q}_i(\boldsymbol{D}_i, \tilde{\boldsymbol{\lambda}}_i, \boldsymbol{\mu}_i, \theta_i)$, or $\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) = \sum_{s=1}^{S}\sum_{j\in\mathcal{J}} c_j b_j(s) + \sum_{i\in\mathcal{I}} \mathcal{Q}_i(\boldsymbol{D}_i, \tilde{\boldsymbol{\lambda}}_i, \boldsymbol{\mu}_i, \theta_i)$, where $\mathcal{Q}_i(\boldsymbol{D}_i, \tilde{\boldsymbol{\lambda}}_i, \boldsymbol{\mu}_i, \theta_i)$ is the waiting and abandonment penalty due to class-i customers, $i \in \mathcal{I}$.

2.2 Data-Driven Approach with Risk Aversion

Data for call centers include arrival data, service rates and abandonment rates. Historical arrival data $\tilde{\boldsymbol{\lambda}}_i \in \mathbb{R}^T$ are collected from K planning intervals in the past, $\boldsymbol{\lambda}_i^1, \dots, \boldsymbol{\lambda}_i^K$, which show the uncertainty and time-varying property of arrival rates. The service rates can be generated from historical call-by-call service time data. According to Gans et al. (2003), many call centers use the *grand* historical averages for service rates. They are usually assumed to be managerial decisions for capacity-planning purposes. In addition, time-varying service rates will significantly affect the tractability of our model. Therefore, we assume that μ_{ij} is set to be constant for all planning periods in this paper. We use the abandonment model discussed by Harrison and Zeevi (2005), which is considered as a standard model in call center modeling. In this model, we assume that there is an exponential distributed random variable τ associated with each customer class i with mean $1/\theta_i$. A class-i customer will abandon the queue if his/her waiting time in the queue exceeds τ units of time. The abandonment rates can then be generated by historical call-by-call waiting times of customers who abandon the queue. Similar to the service rates, we assume that either the averages of all historical abandonment data are used or a managerial decision is made for the values of abandonment rates θ_i, $i \in I$ for all planning intervals. We also assume that the time discretization is fine enough for every customer to stay in the system at least one *unit* period on average, which implies $\mu_{ij} < 1$ and $\theta_i < 1$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$.

Given all historical data, the total cost can be calculated as $\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta})$ for each k, $k = 1, \dots, K$. Traditionally, we solve the problem of minimizing the expected cost

$$\mathbb{E}[\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})] = \frac{1}{K} \sum_{k=1}^{K} \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta})$$

to find solutions for $\boldsymbol{b}$ and $\boldsymbol{D}$. In this paper, we take risk into account and assume that decision makers are risk averse. Consider the set $\mathcal{U}$ of non-decreasing convex disutility functions for risk-averse costs, we would like to find solutions $\boldsymbol{b}$ and $\boldsymbol{D}$ that produce reasonably low expected disutility value $\mathbb{E}[U(\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}))]$ for some $U \in \mathcal{U}$. Bertsimas and Thiele (2006) have applied this approach for newsvendor problems.

Utility theory and stochastic dominance has been studied intensively in economics and finance for expected returns instead of costs. To be simple, we consider the equivalent problem of maximizing the value $\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) = -\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ and the set $\bar{\mathcal{U}}$ of non-decreasing concave utility functions for risk-averse returns. Conversion between the set $\mathcal{U}$ for risk-averse costs and $\bar{\mathcal{U}}$ is straightforward with the notion of negative returns. The main question then becomes finding solutions $\boldsymbol{b}$ and $\boldsymbol{D}$ that produce reasonably high expected utility value $\mathbb{E}[U(\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}))]$ for some $U \in \bar{\mathcal{U}}$.

The function set $\bar{\mathcal{U}}$ relates to the notion of second-order stochastic dominance. The random variable $\tilde{X}$ dominates the random variable $\tilde{Y}$ by second-order stochastic dominance if $\mathbb{E}[U(\tilde{X})] \geq \mathbb{E}[U(\tilde{Y})]$ for all $U \in \bar{\mathcal{U}}$ (with at least one strict inequality). This means we should look for $\boldsymbol{b}$ and $\boldsymbol{D}$ such that the corresponding random variable $\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ likely dominates others by second-order stochastic dominance. Levy (2006) presents various stochastic dominance rules in terms of cumulative distributions and also distribution quantiles. Let $q_\alpha(\tilde{X})$ be the (lower) α-quantile of $\tilde{X}$,

$$q_\alpha(\tilde{X}) := \inf\{x \,|\, \mathbb{P}(\tilde{X} \leq x) \geq \alpha\},\ \alpha \in (0,1),$$

then the second-order stochastic dominance can be characterized as follows:

Theorem 1 (Levy (2006)) *$\tilde{X}$ dominates $\tilde{Y}$ by second-order stochastic dominance if and only if*

$$\int_0^\alpha q_a(\tilde{X})\mathrm{d}a \geq \int_0^\alpha q_a(\tilde{Y})\mathrm{d}a,\ \forall\, \alpha \in (0,1)$$

with at least one strict inequality.

This quantile condition can also be expressed as $ES_\alpha(\tilde{X}) \leq ES_\alpha(\tilde{Y})$ for all $\alpha \in (0,1)$, where the expected shortfall $ES_\alpha(\tilde{X})$ is defined as follows:

$$ES_\alpha(\tilde{X}) := -\frac{1}{\alpha}\int_0^\alpha q_a(\tilde{Y})\mathrm{d}a,\ \alpha \in (0,1).$$

According to Theorem 1, if we choose to minimize the expected shortfall for a fixed value of α, we get a *non-dominated* solution. It means that no other solution can improve the expected utility value of the return for *all* risk-averse decision makers. We also have:

$$\lim_{\alpha \to 0} ES_\alpha(\tilde{X}) = -\inf \tilde{X}, \quad \lim_{\alpha \to 1} ES_\alpha(\tilde{X}) = -\mathbb{E}[\tilde{X}].$$

This shows that if we vary α, the solution will vary form the most conservative (but robust) solution to the solution of the risk-neutral problem.

Applying this approach to our problem, we obtain the following minimization problem:

$$\begin{array}{lll} \min_{\boldsymbol{b},\boldsymbol{D}} & ES_\alpha(\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})) & \\ \text{s.t.} & \sum_{i\in\mathcal{I}} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1,\ldots,T, \\ & d_{ij}(t) \geq 0, & t = 1,\ldots,T, \\ & b_j(s) \geq 0, & s = 1,\ldots,S. \end{array} \tag{6}$$

Given the historical data, the expected shortfall can be estimated non-parametrically. We order the values $\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta})$ in an increasing order,

$$\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^{(k)}, \boldsymbol{M}, \boldsymbol{\theta}) \leq \mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^{(k+1)}, \boldsymbol{M}, \boldsymbol{\theta}), \, \forall k.$$

Define $K_\alpha = \lfloor K\alpha + (1-\alpha\rfloor$ for $\alpha \in [0,1]$. K_α takes all values from 1 to K when α varies from 0 to 1 with $K_0 = 1$ and $K_1 = K$. The expected shortfall is then estimated as follows:

$$ES_\alpha(\mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})) \approx -\frac{1}{K_\alpha} \sum_{k=1}^{K_\alpha} \mathcal{R}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^{(k)}, \boldsymbol{M}, \boldsymbol{\theta}).$$

If we again order $\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta})$ in an increasing order, then the minimization problem defined in (6) is equivalent to the following problem:

$$\begin{array}{lll}
\min_{\boldsymbol{b},\boldsymbol{D}} & \frac{1}{K_\alpha} \sum_{k=K-K_\alpha+1}^{K} \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^{(k)}, \boldsymbol{M}, \boldsymbol{\theta}) & \\
\text{s.t.} & \sum_{i\in\mathcal{I}} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1, \ldots, T, \\
& d_{ij}(t) \geq 0, & t = 1, \ldots, T, \\
& b_j(s) \geq 0, & s = 1, \ldots, S.
\end{array} \tag{7}$$

We are going to prove the following theorem:

Theorem 2 *Problem (7) is equivalent to the following linear program:*

$$\begin{array}{lll}
\min & v + \frac{1}{K_\alpha} \sum_{k=1}^{K} w_k & \\
\text{s.t.} & v + w_k \geq \sum_{s=1}^{S} \sum_{j\in\mathcal{J}} c_j b_j(s) + \sum_{t=1}^{T} \sum_{i\in\mathcal{I}} (k_i^q + k_i^a \theta_i) q_i^k(t), & k = 1, \ldots, K, \\
& q_i^k(t) = (1-\theta_i) q_i^k(t-1) - \sum_{j\in\mathcal{J}} u_{ij}^k(t) + \tilde{\lambda}_i^k(t), & t = 1, \ldots, T, \, k = 1, \ldots, K, \\
& q_i^k(t), u_{ij}^k(t) \geq 0, & t = 1, \ldots, T, \, k = 1, \ldots, K, \\
& s_{ij}^k(t) = (1-\mu_{ij}) s_{ij}^k(t-1) + u_{ij}^k(t), & t = 1, \ldots, T, \, k = 1, \ldots, K, \\
& 0 \leq s_{ij}^k(t) \leq d_{ij}(t), & t = 1, \ldots, T, \, k = 1, \ldots, K, \\
& q_i^k(0) = 0, s_{ij}^k(0) = 0, & i \in \mathcal{I}, \, j \in \mathcal{J}, \, k = 1, \ldots, K, \\
& w_k \geq 0, & k = 1, \ldots, K, \\
& \sum_{i\in\mathcal{I}} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1, \ldots, T, \\
& d_{ij}(t) \geq 0, & t = 1, \ldots, T, \\
& b_j(s) \geq 0, & s = 1, \ldots, S.
\end{array} \tag{8}$$

Proof. We have, the sum $\sum_{k=K-K_\alpha+1}^{K} \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^{(k)}, \boldsymbol{M}, \boldsymbol{\theta})$ can be calculated using the following linear program:

$$\begin{array}{lll}
\max_{\boldsymbol{x}} & \sum_{k=1}^{K} \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta}) x_k & \\
\text{s.t.} & \sum_{k=1}^{K} x_k = K_\alpha, & \\
& 0 \leq x_k \leq 1, & \forall\, k = 1, \ldots, K.
\end{array}$$

Applying strong duality, we can calculate the given sum using the dual problem:

$$\begin{array}{rll} \min_{v,\boldsymbol{w}} & K_\alpha v + \sum_{k=1}^K w_k & \\ \text{s.t.} & v + w_k \geq \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta}), & \forall\, k = 1, \ldots, K, \\ & w_k \geq 0, & \forall\, k = 1, \ldots, K. \end{array}$$

Thus, we can rewrite Problem (7) as follows:

$$\begin{array}{rll} \min & v + \frac{1}{K_\alpha} \sum_{k=1}^K w_k & \\ \text{s.t.} & v + w_k \geq \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta}), & \forall\, k = 1, \ldots, K, \\ & w_k \geq 0, & \forall\, k = 1, \ldots, K, \\ & \sum_{i\in\mathcal{I}} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1, \ldots, T, \\ & d_{ij}(t) \geq 0, & t = 1, \ldots, T, \\ & b_j(s) \geq 0, & s = 1, \ldots, S. \end{array}$$

We have: $\mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta}) = \sum_{j\in\mathcal{J}} c_j b_j + \mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$, where $\mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ is the optimal value of the minimization problem defined in (4). The constraint $v + w_k \geq \mathcal{C}(\boldsymbol{b}, \boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}^k, \boldsymbol{M}, \boldsymbol{\theta})$ will be unchanged if we replace the optimality for $\mathcal{Q}(\boldsymbol{D}, \tilde{\boldsymbol{\Lambda}}, \boldsymbol{M}, \boldsymbol{\theta})$ by its corresponding feasibility due to the nature of the constraint.

Using these arguments, Problem (7) can then be reformulated as follows:

$$\begin{array}{rll} \min & v + \frac{1}{K_\alpha} \sum_{k=1}^K w_k & \\ \text{s.t.} & v + w_k \geq \sum_{s=1}^S \sum_{j\in J} c_j b_j(s) + \sum_{t=1}^T \sum_{i\in I} (k_i^q + k_i^a \theta_i) q_i^k(t), & k = 1, \ldots, K, \\ & q_i^k(t) = (1-\theta_i) q_i^k(t-1) - \sum_{j\in\mathcal{J}} u_{ij}^k(t) + \tilde{\lambda}_i^k(t), & t = 1, \ldots, T,\ k = 1, \ldots, K, \\ & q_i^k(t), u_{ij}^k(t) \geq 0, & t = 1, \ldots, T,\ k = 1, \ldots, K, \\ & s_{ij}^k(t) = (1-\mu_{ij}) s_{ij}^k(t-1) + u_{ij}^k(t), & t = 1, \ldots, T,\ k = 1, \ldots, K, \\ & 0 \leq s_{ij}^k(t) \leq d_{ij}(t), & t = 1, \ldots, T,\ k = 1, \ldots, K, \\ & q_i^k(0) = 0, s_{ij}^k(0) = 0, & i \in \mathcal{I},\ j \in \mathcal{J},\ k = 1, \ldots, K, \\ & w_k \geq 0, & k = 1, \ldots, K, \\ & \sum_{i\in\mathcal{I}} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1, \ldots, T, \\ & d_{ij}(t) \geq 0, & t = 1, \ldots, T, \\ & b_j(s) \geq 0, & s = 1, \ldots, S. \end{array}$$

□

3 Robust Optimization Approach

The robust optimization approach proposed in this section considers all possible worst-case scenarios regarding arrival rates. If arrival rates belong to the uncertainty set U^λ, then we need to solve the following problem:

$$\begin{array}{rlr} \min_{\boldsymbol{b},\boldsymbol{D}} & \sum_{s=1}^{S}\sum_{j\in\mathcal{J}} c_j b_j(s) + \max_{\boldsymbol{\Lambda}\in U^\lambda} \mathcal{Q}(\boldsymbol{D},\boldsymbol{\Lambda},\boldsymbol{M},\boldsymbol{\theta}) & \\ \text{s.t.} & \sum_{i\in\mathcal{I}} d_{ij}(t) \le b_j(\lceil tS/T\rceil), & t=1,\ldots,T, \\ & d_{ij}(t)\ge 0, & t=1,\ldots,T, \\ & b_j(s)\ge 0, & s=1,\ldots,S. \end{array} \tag{9}$$

If we assume the *separability* of the uncertainty set U^λ with respect to customer classes, $U^\lambda = \prod_{i\in\mathcal{I}} U_i^\lambda$ where $U_i^\lambda \subset \mathbb{R}_+^T$ for all $i\in\mathcal{I}$, then the robust formulation can be written as follows:

$$\begin{array}{rlr} \min_{\boldsymbol{b},\boldsymbol{D}} & \sum_{s=1}^{S}\sum_{j\in\mathcal{J}} c_j b_j(s) + \sum_{i\in\mathcal{I}} \max_{\boldsymbol{\lambda}_i\in U_i^\lambda} \mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i) & \\ \text{s.t.} & \sum_{i\in\mathcal{I}} d_{ij}(t) \le b_j(\lceil tS/T\rceil), & t=1,\ldots,T, \\ & d_{ij}(t)\ge 0, & t=1,\ldots,T, \\ & b_j(s)\ge 0, & s=1,\ldots,S. \end{array} \tag{10}$$

To analyze the above robust formulation, we focus on properties of $\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i)$, which is the optimal value of the following optimization problem:

$$\begin{array}{rlr} \min_{\boldsymbol{q}_i,\boldsymbol{S}_i,\boldsymbol{U}_i} & \sum_{t=1}^{T}(k_i^q + k_i^a\theta_i)q_i(t) & \\ \text{s.t.} & q_i(t) = (1-\theta_i)q_i(t-1) - \sum_{j\in\mathcal{J}} u_{ij}(t) + \lambda_i(t), & t=1,\ldots,T, \\ & q_i(t), u_{ij}(t)\ge 0, & t=1,\ldots,T, \\ & s_{ij}(t) = (1-\mu_{ij})s_{ij}(t-1) + u_{ij}(t), & t=1,\ldots,T, \\ & 0\le s_{ij}(t)\le d_{ij}(t), & t=1,\ldots,T, \\ & q_i(0)=0, s_{ij}(0)=0, & j\in\mathcal{J}. \end{array} \tag{11}$$

We will consider $\boldsymbol{D}_i$ such that the above optimization problem is feasible. The following proposition shows that $\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i)$ increases in $\lambda_i(t)$:

Proposition 1 *Let $\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i)$ be the optimal objective of problem defined in (11). For any $t=1,\ldots,T$ and $\delta>0$,*

$$\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i+\delta\boldsymbol{e}(t),\boldsymbol{\mu}_i,\theta_i) \ge \mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i),$$

where $\boldsymbol{e}(t)$ is the t-th unit vector in $\mathbb{R}^T$.

Proof. Consider the modified optimization problem

$$\begin{aligned}\mathcal{Q}_i'(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i) = \min_{\boldsymbol{q}_i,\boldsymbol{S}_i,\boldsymbol{U}_i} \quad & \textstyle\sum_{t=1}^{T}(k_i^q + k_i^a\theta_i)q_i(t) & \\ \text{s.t.} \quad & q_i(t) \geq (1-\theta_i)q_i(t-1) - \textstyle\sum_{j\in\mathcal{J}} u_{ij}(t) + \lambda_i(t), & t = 1,\ldots,T, \\ & q_i(t), u_{ij}(t) \geq 0, & t = 1,\ldots,T, \\ & s_{ij}(t) = (1-\mu_{ij})s_{ij}(t-1) + u_{ij}(t), & t = 1,\ldots,T, \\ & 0 \leq s_{ij}(t) \leq d_{ij}(t), & t = 1,\ldots,T, \\ & q_i(0) = 0, s_{ij}(0) = 0, & j \in \mathcal{J}.\end{aligned} \tag{12}$$

We will prove that $\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i) = \mathcal{Q}_i'(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i)$. Let $(q_i^*(t), s_{ij}^*(t), u_{ij}^*(t))$ be an optimal solution of the problem defined in (11). Clearly, $(q_i^*(t), s_{ij}^*(t), u_{ij}^*(t))$ is a feasible solution for (12). Thus

$$\mathcal{Q}_i(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i) \geq \mathcal{Q}_i'(\boldsymbol{D}_i,\boldsymbol{\lambda}_i,\boldsymbol{\mu}_i,\theta_i).$$

Now consider an optimal solution $(q_i'(t), s_{ij}'(t), u_{ij}'(t))$ of the problem defined in (12). We will prove that if $q_i'(t+1) > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$ for some $t = 0,\ldots,T-1$ then $q_i'(t+1) = 0$. Assume that there exists t such that $q_i'(t+1) > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$ and $q_i'(t+1) > 0$. Replacing $q_i'(t+1)$ by $q_i'(t+1) - \epsilon$ for some $0 < \epsilon < q_i'(t+1)$ such that

$$q_i'(t+1) - \epsilon > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1).$$

If $t < T-1$, we have:

$$q_i'(t+2) \geq (1-\theta_i)q_i'(t+1) - \sum_{j\in\mathcal{J}} u_{ij}'(t+2) + \lambda_i(t+2) > (1-\theta_i)(q_i'(t+1)-\epsilon) - \sum_{j\in\mathcal{J}} u_{ij}'(t+2) + \lambda_i(t+2).$$

Thus the new solution is feasible with lower cost as $k_i^q + \theta_i k_i^q > 0$ (contradiction). This implies that if $q_i'(t+1) > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$ for some $t = 0,\ldots,T-1$ then $q_i'(t+1) = 0$.

Now consider an optimal solution $(q_i'(t), s_{ij}'(t), u_{ij}'(t))$ such that $q_i'(t+1) > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$ and $q_i'(t+1) = 0$ for some $t = 0,\ldots,T-1$. We will construct another optimal solution $(q_i''(t), s_{ij}''(t), u_{ij}''(t))$ in which additional equality $q_i''(t+1) = (1-\theta_i)q_i''(t) - \sum_{j\in\mathcal{J}} u_{ij}''(t+1) + \lambda_i(t+1)$ is obtained. We have: $(1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda(t+1) < 0$, thus $\sum_{j\in\mathcal{J}} u_{ij}'(t+1) > (1-\theta_i)q_i'(t) + \lambda_i(t+1) \geq 0$. Let $\Delta = \sum_{j\in\mathcal{J}} u_{ij}'(t+1) - [(1-\theta_i)q_i'(t) + \lambda(t+1)] > 0$ and define

$$u_{ij}''(t+1) = u_{ij}'(t+1) - \frac{u_{ij}'(t+1)}{\sum_{k\in\mathcal{J}} u_{ik}'(t+1)}\Delta, \quad \forall\, j \in \mathcal{J}.$$

We have: $0 \leq u_{ij}''(t+1) \leq u_{ij}'(t+1)$ for all $j \in \mathcal{J}$ and if we define $s_{ij}''(t+1) = (1-\mu_{ij})s_{ij}'(t) + u_{ij}''(t+1)$, we then have $0 \leq s_{ij}''(t+1) \leq s_{ij}'(t+1)$. Similarly, let $s_{ij}''(\tau+1) = (1-\mu_{ij})s_{ij}''(\tau) + u_{ij}'(\tau+1) \leq s_{ij}'(\tau+1)$

for all $\tau \geq t+1$, we maintain the problem feasibility while keeping other solution values. We can repeat this procedure for all t such that $q_i'(t+1) > (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$. Thus there exists an optimal solution $(q_i'(t), s_{ij}'(t), u_{ij}'(t))$ such that $q_i'(t+1) = (1-\theta_i)q_i'(t) - \sum_{j\in\mathcal{J}} u_{ij}'(t+1) + \lambda_i(t+1)$ for all $t = 0, \ldots, T-1$. This implies that it is a feasible solution for problem defined in (11). Therefore, we have:

$$\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) \leq \mathcal{Q}_i'(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i).$$

From these two results, we obtain $\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) = \mathcal{Q}_i'(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i)$.

We only consider $\boldsymbol{D}_i$ such that (11) is feasible. In addition, $k_i^q + \theta_i k_i^a > 0$ and $q_i(t) \geq 0$ for all $t = 1, \ldots, T$; therefore, $0 \leq \mathcal{Q}_i'(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) < +\infty$. This implies that the dual problem is feasible and strong duality holds. Let $\pi_i(t) \geq 0$, $p_{ij}(t) \leq 0$, and $r_{ij}(t)$, $t = 1, \ldots, T$, $j \in \mathcal{J}$, be the dual variables with respect to the set of constraints $q_i(t) \geq (1-\theta_i)q_i(t-1) - \sum_{j\in\mathcal{J}} u_{ij}(t) + \lambda_i(t)$, $s_{ij}(t) \leq d_{ij}(t)$, and $s_{ij}(t) = (1-\mu_{ij})s_{ij}(t-1) + u_{ij}(t)$ respectively, the dual problem is formulated as follows:

$$\begin{array}{lll}
\max & \sum_{t=1}^T \pi_i(t)\lambda_i(t) + \sum_{t=1}^T \sum_{j\in\mathcal{J}} p_{ij}(t) d_{ij}(t) & \\
\text{s.t.} & \pi_i(t) - (1-\theta_i)\pi_i(t+1) \leq k_i^q + \theta_i k_i^a, & t = 1, \ldots, T, \\
& \pi_i(t) - r_{ij}(t) \leq 0, & j \in \mathcal{J},\ t = 1, \ldots, T, \\
& r_{ij}(t) - (1-\mu_{ij}) r_{ij}(t+1) + p_{ij}(t) \leq 0, & j \in \mathcal{J},\ t = 1, \ldots, T, \\
& \pi_i(T+1) = 0,\ p_{ij}(T+1) = 0, & j \in \mathcal{J},\ t = 1, \ldots, T, \\
& \pi_i(t) \geq 0,\ p_{ij}(t) \leq 0, & j \in \mathcal{J},\ t = 1, \ldots, T.
\end{array}$$

We have: $r_{ij}(t) \geq \pi_i(t) \geq 0$ and $p_{ij}(t) \leq 0$ for all $j \in \mathcal{J}$, $t = 1, \ldots, T$, thus the dual problem has extreme points. Therefore, we have:

$$\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) = \max_{k=1,\ldots,K} \sum_{t=1}^T \pi_i^k(t)\lambda_i(t) + \sum_{t=1}^T \sum_{j\in\mathcal{J}} p_{ij}^k(t) d_{ij}(t),$$

where $(\pi_i^k(t), p_{ij}^k(t), r_{ij}^k(t))$ are dual extreme points, $k = 1, \ldots, K$.

Assume that $\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) = \sum_{t=1}^T \pi_i^{\bar{k}}(t)\lambda_i(t) + \sum_{t=1}^T \sum_{j\in\mathcal{J}} p_{ij}^{\bar{k}}(t) d_{ij}(t)$ for some $\bar{k}$, we then have:

$$\begin{aligned}
\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i + \delta \boldsymbol{e}(t), \boldsymbol{\mu}_i, \theta_i) &= \max_{k=1,\ldots,K} \sum_{\tau=1}^T \pi_i^k(\tau)\lambda_i(\tau) + \sum_{\tau=1}^T \sum_{j\in\mathcal{J}} p_{ij}^k(\tau) d_{ij}(\tau) + \delta \pi_i^k(t) \\
&\geq \sum_{\tau=1}^T \pi_i^{\bar{k}}(\tau)\lambda_i(\tau) + \sum_{\tau=1}^T \sum_{j\in\mathcal{J}} p_{ij}^{\bar{k}}(\tau) d_{ij}(\tau) + \delta \pi_i^{\bar{k}}(t) \\
&\geq \mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i)
\end{aligned}$$

Thus we have: $\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i + \delta \boldsymbol{e}(t), \boldsymbol{\mu}_i, \theta_i) \geq \mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i)$ for all $\delta > 0$ and $t = 1, \ldots, T$. □

This property of $\mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i)$ shows if $U_i^\lambda = \left\{\boldsymbol{\lambda}_i \in \mathbb{R}_+^T : \lambda_i(t) \leq \bar{\lambda}_i(t),\, \forall\, t = 1, \ldots, T\right\}$ for some $\bar{\boldsymbol{\lambda}}_i \in \mathbb{R}_+^T$, then

$$\max_{\boldsymbol{\lambda}_i \in U_i^\lambda} \mathcal{Q}_i(\boldsymbol{D}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \theta_i) = \mathcal{Q}_i(\boldsymbol{D}_i, \bar{\boldsymbol{\lambda}}_i, \boldsymbol{\mu}_i, \theta_i). \tag{13}$$

Given historical data $\boldsymbol{\lambda}_i^k$, $k = 1, \ldots, K$, we can calculate the mean arrival rates $\bar{\boldsymbol{\lambda}}_i(t)$ and its standard deviation $\sigma_i^\lambda(t)$ for all $t = 1, \ldots, T$. Define the uncertainty sets with the parameter $\Gamma \geq 0$ as follows:

$$U_i^\lambda(\Gamma) = \left\{\boldsymbol{\lambda}_i \in \mathbb{R}_+^T : \lambda_i(t) \leq \bar{\lambda}_i(t) + \Gamma\sigma_i^\lambda(t),\, \forall\, t = 1, \ldots, T\right\}. \tag{14}$$

The following theorem shows the robust formulation with these uncertainty sets:

Theorem 3 *The robust formulation (9) with uncertainty set $U^\lambda(\Gamma) = \prod_{i \in \mathcal{I}} U_i^\lambda(\Gamma)$, where $U_i^\lambda(\Gamma)$ is defined in (14), is equivalent to the following linear programming:*

$$\begin{array}{lll}
\min & \sum_{s=1}^S \sum_{j \in \mathcal{J}} c_j b_j(s) + \sum_{t=1}^T \sum_{i \in \mathcal{I}} (k_i^q + k_i^a \theta_i) q_i(t) & \\
\text{s.t.} & q_i(t) = (1 - \theta_i) q_i(t-1) - \sum_{j \in \mathcal{J}} u_{ij}(t) + \bar{\lambda}_i(t) + \Gamma\sigma_i^\lambda(t), & t = 1, \ldots, T, \\
& q_i(t), u_{ij}(t) \geq 0, & t = 1, \ldots, T, \\
& s_{ij}(t) = (1 - \mu_{ij}) s_{ij}(t-1) + u_{ij}(t), & t = 1, \ldots, T, \\
& 0 \leq s_{ij}(t) \leq d_{ij}(t), & t = 1, \ldots, T, \\
& q_i(0) = 0, s_{ij}(0) = 0, & i \in \mathcal{I},\, j \in \mathcal{J}, \\
& \sum_{i \in I} d_{ij}(t) \leq b_j(\lceil tS/T \rceil), & t = 1, \ldots, T, \\
& d_{ij}(t) \geq 0, & t = 1, \ldots, T, \\
& b_j(s) \geq 0, & s = 1, \ldots, S.
\end{array} \tag{15}$$

Proof. Apply the result shown in (13) for the uncertainty set $U_i^\lambda(\Gamma)$ for all $i \in \mathcal{I}$. □

4 Computational Results

4.1 Dynamic Routing Policy

The optimal *off-line* customer-agent allocation for the whole planning interval, $d_{ij}(t)$, $t = 1, \ldots, T$, obtained from either data-driven or robust optimization approach, allows us to generate a dynamic routing policy for the call center system. There are two main cases in which a routing decision needs to be made. The first case is when a customer arrives and there are more than one agent pool with available agents that can serve that customer. The second case is when an agent finishes serving a customer and there is more than one customer queue from which customers can be served by the agent.

At each time t, the number of class-i customers being served by agents in pool j, $s_{ij}(t)$, is known. If a class-i customer arrives at time t, let $\mathcal{J}_i(t) = \{j \in \mathcal{J} : \mu_{ij} > 0, s_{ij}(t) \leq b_j - 1\}$. A routing decision needs to be made when $|\mathcal{J}_i(t)| > 1$. The selected agent pool belongs to the set

$$\arg\min_{j \in \mathcal{J}_i(t)} \left\{ s_{ij}(t) - \frac{d_{ij}(t)}{\sum_{k \in \mathcal{J}_i(t)} d_{ik}(t)} \sum_{k \in \mathcal{J}_i(t)} s_{ik}(t) \right\},$$

where ties are broken by (faster) service rate and arbitrarily afterwards.

Similarly, if an agent in pool j finishes serving a customer, let $\mathcal{I}_j(t) = \{i \in \mathcal{I} : \mu_{ij} > 0, q_i(t) > 0\}$, where $q_i(t)$ is the number of class-i customers waiting in queue at time t. A routing decision needs to be made when $|\mathcal{I}_j(t)| > 1$. The selected customer class belongs to the set

$$\arg\min_{i \in \mathcal{I}_j(t)} \left\{ s_{ij}(t) - \frac{d_{ij}(t)}{\sum_{k \in \mathcal{I}_j(t)} d_{kj}(t)} \sum_{k \in \mathcal{I}_j(t)} s_{kj}(t) \right\},$$

where ties are broken by (higher) staff cost and arbitrarily afterwards. This policy clearly helps us to maintain the predefined *off-line* customer-agent allocation.

In addition, when additional agents are added at the beginning of a new shift, a routing decision needs to be made if there are customers waiting in queue at that time. The waiting customers are ranked according their waiting penalty. One by one, these ranked customers will be routed according to the policy set out in the first case until there is no more possible customer-agent match.

4.2 System Simulation

In this paper, we obtain historical arrival data from a US bank study using the SEESTAT software. Given an arrival rate sample path, the arrival process is assumed to be a non-homogeneous Poisson process, which can be simulated by the thinning procedure described in Ross (1997). Service times are exponentially distributed as usual. Each abandonment rate θ is associated with maximum waiting time of customers before abandoning the call, which is exponentially distributed with mean $1/\theta$ as mentioned in Section 2.2. The call center system with its staff capacity information and the *off-line* customer-agent allocation is then simulated using a discrete event-based simulation. The dynamic routing policy is implemented as described in the previous section. Simulation results are used to measure the system performance for different staffing and customer-agent allocation settings.

The two proposed models, either data-driven or robust optimization one, are parameterized models. The data-driven approach is parameterized by α, the quantile, while the robust optimization one by Γ, the robust degree. We adopt training, validation, and testing approach to both models by dividing the historical data set into three subsets, namely training, validation, and testing subsets. We use the training subset to construct optimal solutions with respect to different values of the model parameter. The validation set is for selecting the best parameter value via simulation and finally, we use the testing set to measure the performance of the selected models.

Using the SEESTAT software, we can extract arrival data, service times, and abandonment rates for different customer classes from a US bank study. There are several customer classes and we focus on the six classes with highest arrival rates. They are retailer (class 1), premium (retailer) (2), business (3), customer loans (4), telesales (5), and online banking (6). We also consider only five agent pools, which correspond to retailer (pool 1), business (2), customer loans (3), telesales (4), and online banking (5) customers. We will use these customer classes and agent pools in different network designs which are discussed in later sections.

There are approximately $N = 300$ weekdays which are recorded and we will use all of these records to generate historical data. The planning interval is set to be from 6:00 to 18:00, which has 720 minutes ($T = 720$). There are 24 30-minute shifts ($S = 24$) to be considered. The mean arrival rates of these six customer classes are plotted in Figure 1. We also show an arrival rate sample of retailer customers, the customer class with highest arrival rate, in the same figure. On the other hand, average service times and abandonment rates are shown in Table 1. If a customer is served by an agent from different pools, we will assume that the average service time is increased but at most 10%.

Customer class	Retailer	Premium	Business	Loans	Telesales	Banking
Service time (seconds)	225.86	283.43	224.70	256.14	379.32	389.19
Abandonment rate (%)	0.51	0.33	1.15	1.17	2.24	0.54

Table 1: Average service times and abandonment rates of six customer classes

In order to construct the training, validation, and testing sets, we generate a random permutation and select 150 samples for the training set while each of validation and testing set has 75 samples. We code both data-driven and robust models in C and solved with CPLEX 9.1 solver using a Unix machine of 4×3GHz and 4×2GB RAM. Due to the sparse structure of the two linear optimization formulations, we use barrier method as the solving method for both models, which saves a very significant amount of computation time over the default dual-simplex method. We solve the data-driven model with values of the quantile α between 0.0 and 1.0 with the increments of 0.1. The maximum robust degree Γ is set to be 3.0 and same increments of 0.1 are applied.

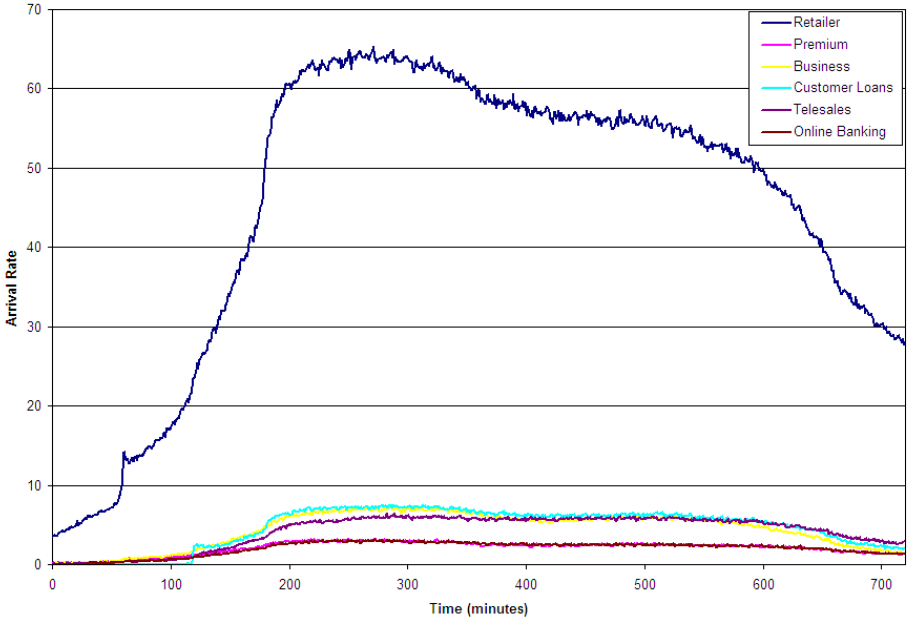

Figure 1: Average arrival rates of six customer classes obtained from the US bank study

4.3 Computational Results

We apply the two proposed models for some canonical network designs presented in Gans et al. (2003), starting with the simplest design I with one customer class and one agent pool to more complicated designs, V, N, and W, which are shown in Figure 2. We will also consider complicated designs which consist of up to six customer classes and five agent pools mentioned in the previous section.

4.3.1 Network Design I

We choose the retailer customer class, which has the highest arrival rate, to work with the simple network design I. Figure 3 shows one random path of arrival data of this customer class. We set the staff cost to be 0.50 per agent per unit time while waiting and abandonment penalty are set to be 1.00 per customer per unit time and 2.00 per customer respectively.

In order to compare computational results of two models, we plot mean-standard deviation frontiers of total costs with respect to different parameters, quantiles and robust degrees for data-driven and robust model respectively, using validation arrival data in Figure 4. For both models, the average total costs increase when the conservative level increases (decrease of the quantile or increase of the robust degree) while the cost standard deviation decreases. The data-driven model yields the best solutions with α between 0.7 and 1.0 with the average total cost of $96,000.00$ and the cost standard deviation of

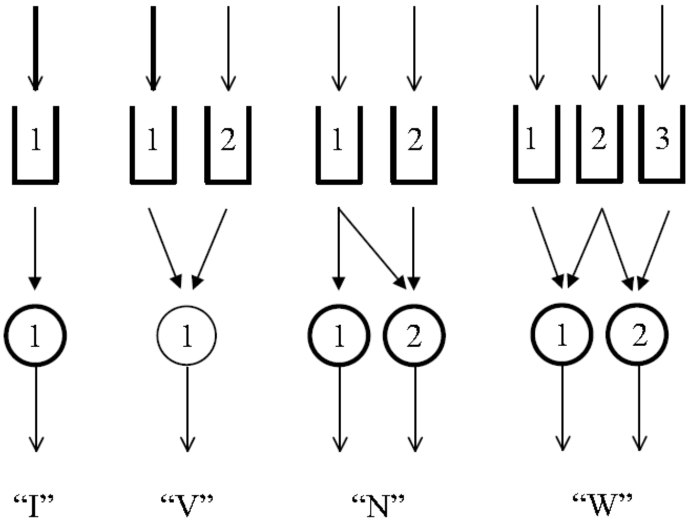

Figure 2: Simple network designs *I*, *V*, *N*, and *W*

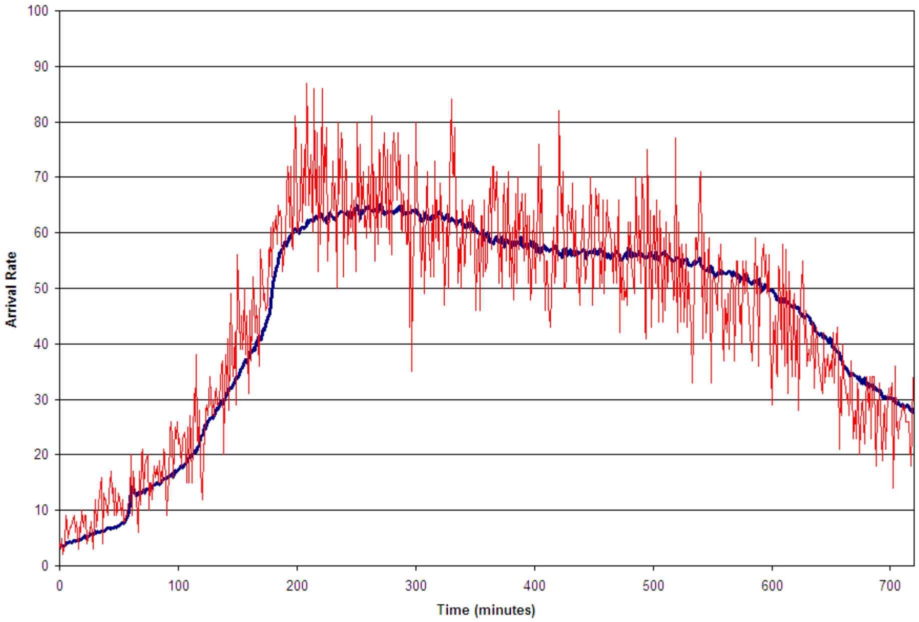

Figure 3: A random arrival sample path of retailer customer class used with the network design *I*

2,000.00 approximately. The optimal cost obtained from the robust model is 89,522.24 when $\Gamma = 1.5$. However, the cost standard deviation is significantly high with this solution, 10,068.96. According to Figure 4, we can increase Γ to obtain better solution than the one from data-driven model in terms of both average total cost and the cost standard deviation. For example, if $\Gamma = 1.8$, we get about 5% decrease in average cost while the cost standard deviation is the same as that of the data-driven model. If we want solutions with smaller cost standard deviation, clearly, the solutions obtained from robust model are also better than those from the data-driven model (smaller average cost with smaller standard deviation).

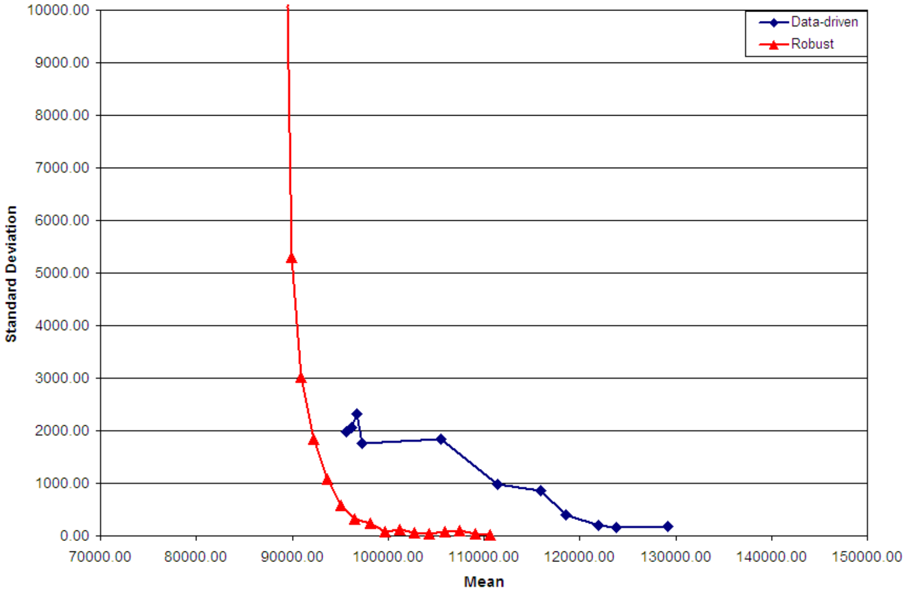

Figure 4: Mean-standard deviation frontiers of two models for network design *I*

For this simple network design, the routing policy is simply first-in-first-out and the staff capacity is the only factor that determines the system performance. We plot here the numbers of agents obtained from the data-driven model with $\alpha = 0.9$ and robust model with $\Gamma = 1.8$ in Figure 5. The graph shows that robust solution requires more staff in the afternoon while the data-driven solution requires more staff around noon. Using these solutions with testing data, we get the average total cost and the cost standard deviation of 95,756.31 and 433.52 for data-driven solution while those of robust solution are 91,984.97 and 1,353.62. The data-driven solution obtains smaller standard deviation, which can be explained probably by the similarity in the arrival trends between training and testing data sets. However, robust solution still has smaller average total cost as in the case with validation arrival data.

This means that arrival uncertainty can be captured well on average with this simple robust model. Another advantage of the robust model is the computational time. We record computational times for both data-driven and robust models and results are shown in Table 2. Clearly, there are significant differences in computational times between data-driven and robust model with this network design I.

Model	I	V	N	W	$C1$	$C2$	$C3$
Data-driven	99.89	1139.02	3272.33	$12,896.35$	$26,866.48$	$28,692.35$	$44,469.47$
Robust	0.08	0.22	0.46	0.68	1.51	2.93	3.45

Table 2: Average computational times (in seconds) for both models with different network designs

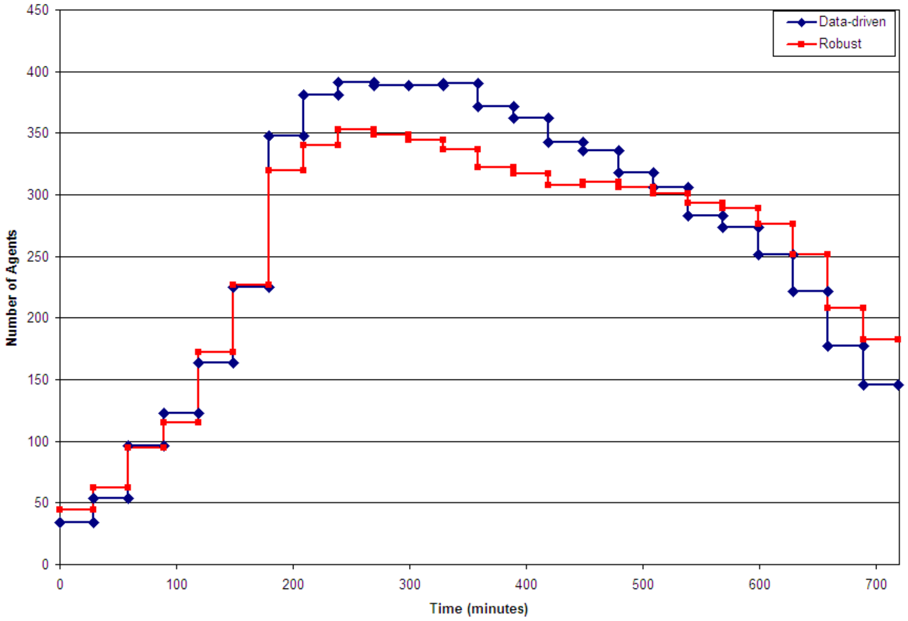

Figure 5: Number of agents in 24 shifts obtained from both models for the network design I

4.3.2 Network Design V

We consider the premier retailer customer class as the second class in the network design V. Due to the importance of this customer class, we set waiting and abandonment penalty to 2.00 and 5.00 respectively. The training, validation, and testing sets are again generated randomly. Similar to the case of the network design I, the mean-standard deviation frontiers are plotted in Figure 6 to evaluate the performance of the two models.

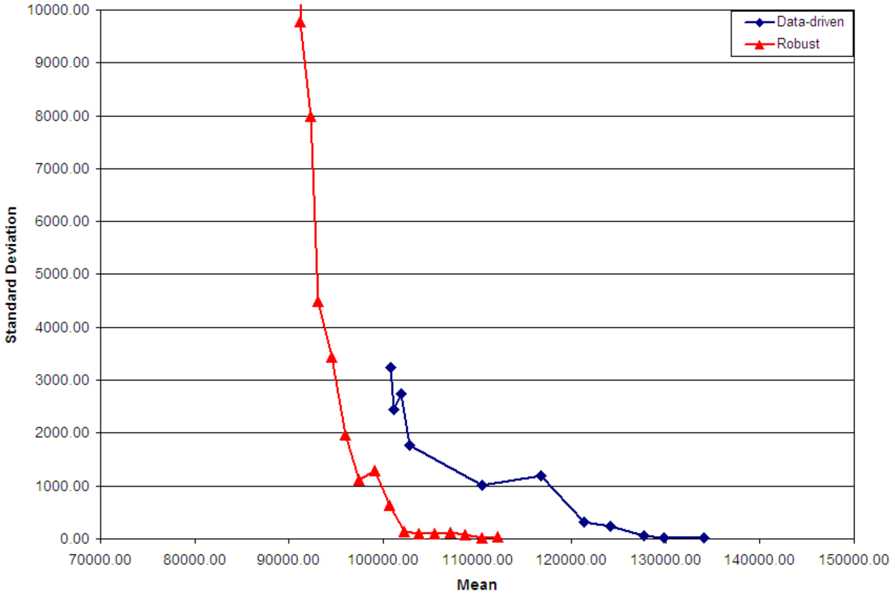

Figure 6: Mean-standard deviation frontiers of two models for the network design V

These computational results show that in terms of average total cost, the robust solution is better than the data-driven one, 91,176.95 with $\Gamma = 1.6$ versus 100,780.79 with $\alpha = 1.0$. However, the cost standard deviation obtained from this robust solution is much higher (9,789.45) than that of the data-driven solution (3,245.26). Similar to the previous case, if we increase the robust degree, we still get smaller average total cost while the cost standard deviation is approximately that of the best data-driven solution. According to Figure 6, if $\Gamma = 1.9$, the average total cost is 94,558.04 and the cost standard deviation is 3,443.73. Using this robust solution with testing data, we again get smaller average cost (97,781.88 as compared to 101,250.17) but higher standard deviation (27,005.29 versus 4,456.55). According to computational results, all testing data get total waiting and abandonment penalty of less than 30,000.00 if the data-driven solution is used. On the other hand, the robust solution results in significantly higher penalty for two arrival paths, approximately 45,000.00 and 240,000.00, which explains why the cost standard deviation is much higher. The result shows that the data-driven solution is probably more tailored to the testing data. This becomes more important as the routing policy is now more complicated than just the simple first-in-first-out policy, which depends on the arrival patterns. Having said that, we still have smaller average total cost when using the robust solution and the computational time is also much smaller than that of the data-driven solution, 0.22 seconds as compared to 1136.02 seconds (see Table 2), which is a significant gain in computational time.

4.3.3 Network Design N

We now assume that premier retailer customers can be served by agents from another agent pool, the business agent pool. The staff cost for this agent pool is 0.40 per agent per unit time, which is lower than the cost of retailer agents. The mean-standard deviation frontiers obtained from two solutions are plotted in Figure 7.

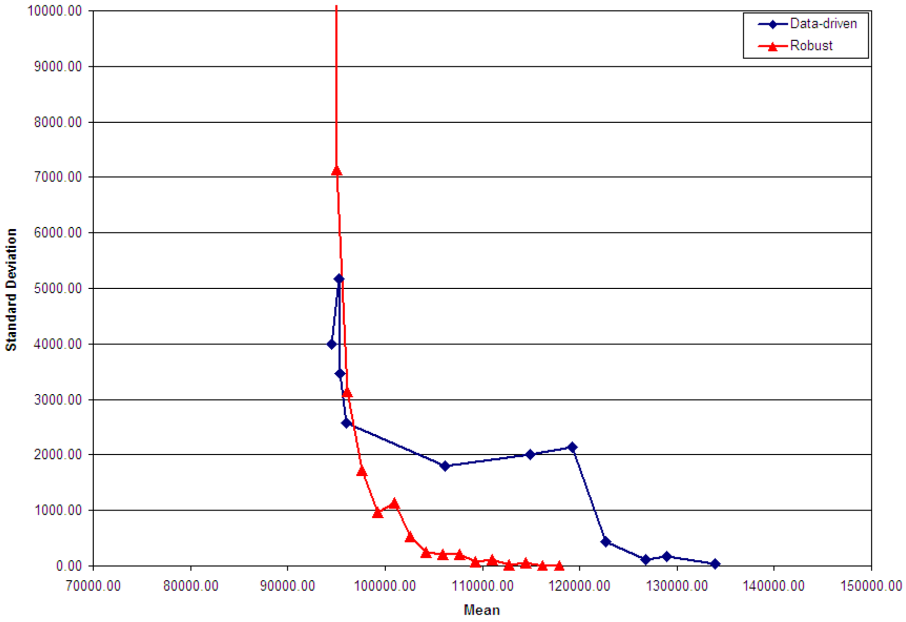

Figure 7: Mean-standard deviation frontiers of two models for the network design N

According to Figure 7, the data-driven model obtains the best average total costs when α between 0.7 and 1.0 with reasonable cost standard deviation. The robust model with $\Gamma = 1.7$ has the slightly higher average total cost but smaller cost standard deviation. If we need solutions with smaller standard deviation, the robust model again provides better solutions than the data-driven counterpart.

Using $\alpha = 0.8$ and $\Gamma = 1.7$ as selected parameters for two models, we examine the solutions by plotting numbers of agents of the second agent pool in Figure 8. The graph shows that the second agent pool is used more under the robust model than the data-driven one. The latter solution is more fluctuating, probably due to the fact that data-driven approach is more data-dependent than the robust one.

We now test two solutions with the testing data set. The results of average total cost and standard deviation are $(103,650.96; 71,744.74)$ and $(106,705.80; 91,980.43)$ for the data-driven and robust solution respectively. As compared to results from validation arrival data, the standard deviations from

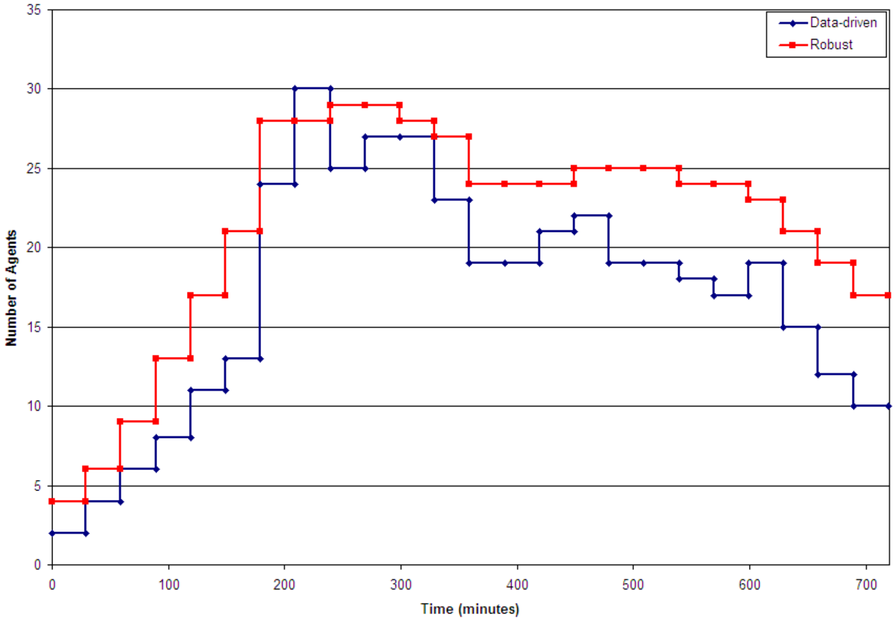

Figure 8: Number of agents in 24 shifts obtained from both models for the second agent pool

testing data are much higher. The reason is that there is a single arrival sample with significantly high arrival rate (see Figure 9), which results in more than 600,000.00 of waiting and abandonment penalty while the average total penalty is no more than 10,000.00. If this arrival sample is removed from the testing set, the cost and standard deviation results are (95,373.31; 2,909.75) and (96,087.75; 2,176.06) respectively. The robust solution yields higher average total cost but smaller cost standard deviation. The result also shows that both models do not perform well if arrival data change significantly from the past data. If more conservative solutions (higher average cost with smaller standard deviation) are needed to encounter these exceptions, the robust model probably can provide better solutions according the mean-standard deviation frontiers. The computational time is again another advantage of the robust model over the data-driven model (see Table 2).

4.3.4 Network Design *W*

In order to build the network design W, we add the business customer class into the existing system. The waiting and abandonment penalty are set to be 2.00 and 4.00 respectively for this customer class. The mean-standard deviation frontiers of two models are plotted in Figure 10.

Under this network design, we again get the best data-driven solutions when α is between 0.7 and 1.0. The best robust solution with similar cost standard deviation is the solution with $\Gamma = 1.9$, which also

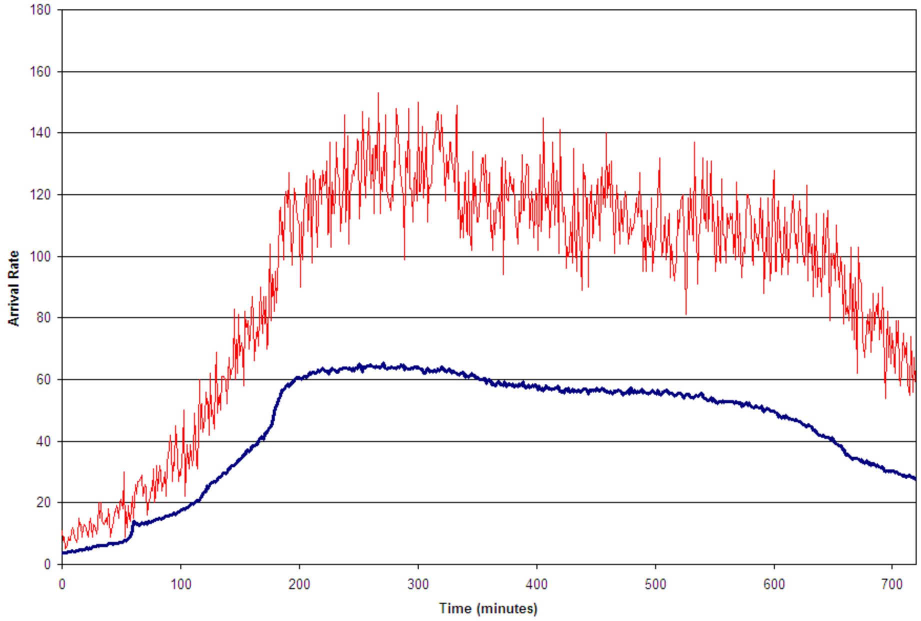

Figure 9: The arrival sample path of retailer customer class which yields the highest waiting and abandonment penalty

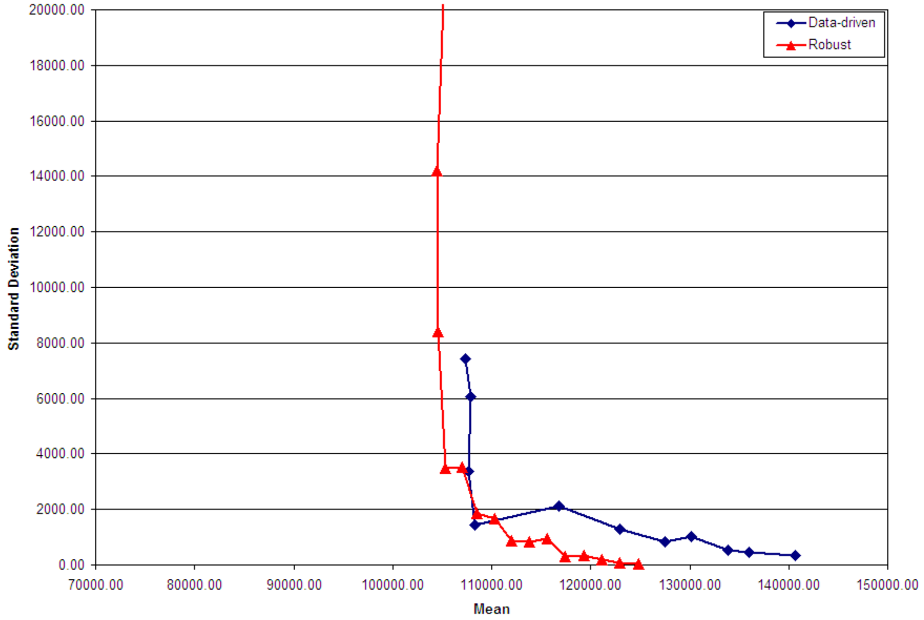

Figure 10: Mean-standard deviation frontiers of two models for the network design W

yields similar average total cost. If we would like to consider more conservative solutions, we can select, for example, $\alpha = 0.7$ and $\Gamma = 2.1$, with which the cost standard deviations decrease while the average total costs slightly increase. Using these solutions with testing data, we obtain the results of average total cost and cost standard deviation of $(108,572.07; 2,236.25)$ and $(109,053.97; 5,353.57)$ for data-driven and robust model respectively. The results show that the data-driven solution is slightly better than the robust counterpart with this network design if we select model parameters as above. This is also due to the fact that the routing policy is more complicated with this network design and it depends greatly on actual arrival data, which gives the data-driven approach an advantage. However, similar to other cases, the robust approach again has a significant advantage with respect to computational time, especially when the network design is bigger (see Table 2).

4.3.5 Complicated Network Designs

In this section, we work with three network designs which are more complicated. The network designs $C1$, $C2$, and $C3$ are shown in Figure 11 and 12. The largest network $C3$ consists of all six customer classes and five agent pools. The cost parameters are written in Table 3.

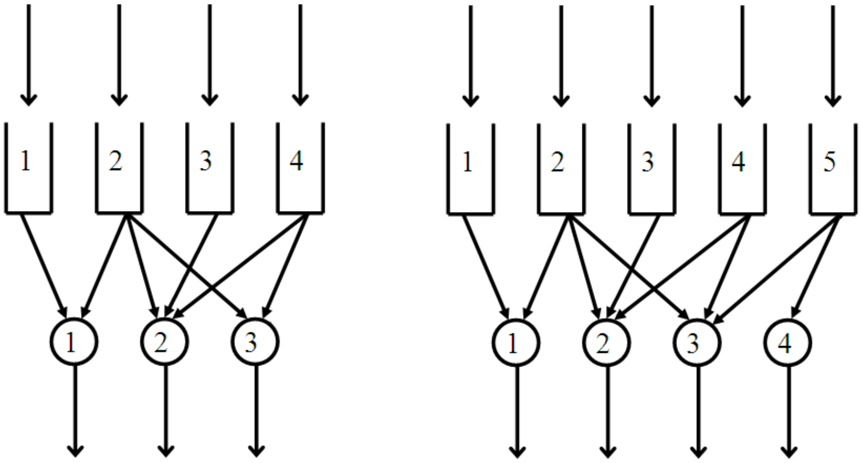

Figure 11: Network design $C1$ and $C2$

The mean-standard deviation frontiers of both models for these three network designs are plotted in Figure 13, 14, and 15 respectively. The results show that the robust model performs as well as the data-drive model in most cases, especially when conservative solutions are needed. The robust solution generates smaller variation in the total cost. In the case when the data-driven model is better

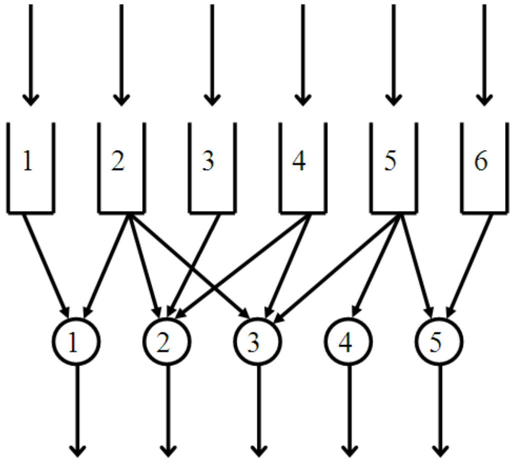

Figure 12: Network design $C3$

Customer class	Retailer	Premium	Business	Loans	Telesales	Banking
Waiting penalty	1.00	2.00	2.00	1.50	1.50	1.50
Abandonment penalty	2.00	5.00	4.00	4.00	3.00	3.00
Agent pool	Retailer	Business	Loans	Telesales	Banking	-
Staff cost	0.50	0.60	0.60	0.50	0.60	-

Table 3: Customer waiting and abandonment penalties and agent personnel costs

in terms of cost expectation, the relative difference is small (5% for $C3$) as compared to the difference in computational time. Table 2 shows this huge difference for all three network designs. For the network design $C3$, the robust approach needs less than 4 seconds while the data-driven approach takes more than 12 hours on average.

We have presented computational results for different network design, from simple canonical designs to complicated ones with data obtained from a US bank study using the training, validation, and testing approach. In terms of average total cost, the data-driven model obtains the best solution with reasonable cost standard deviation when α between 0.8 and 1.0 with these data from the study. Similarly, the suitable value of Γ for the robust model is around 2.0. The robust model outperforms or at least performs as well as the data-driven counterpart for most of the network designs, especially if we want more conservative solutions. In some cases, the data-driven approach is slightly better with less than 5% improvement in total cost with the same cost variation as compared to the robust approach. In

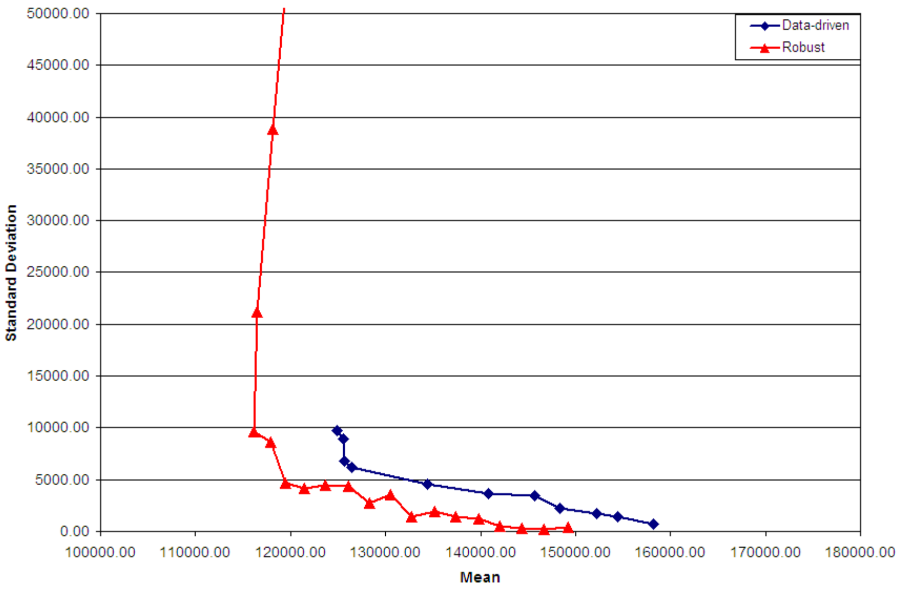

Figure 13: Mean-standard deviation frontiers of two models for the network design $C1$

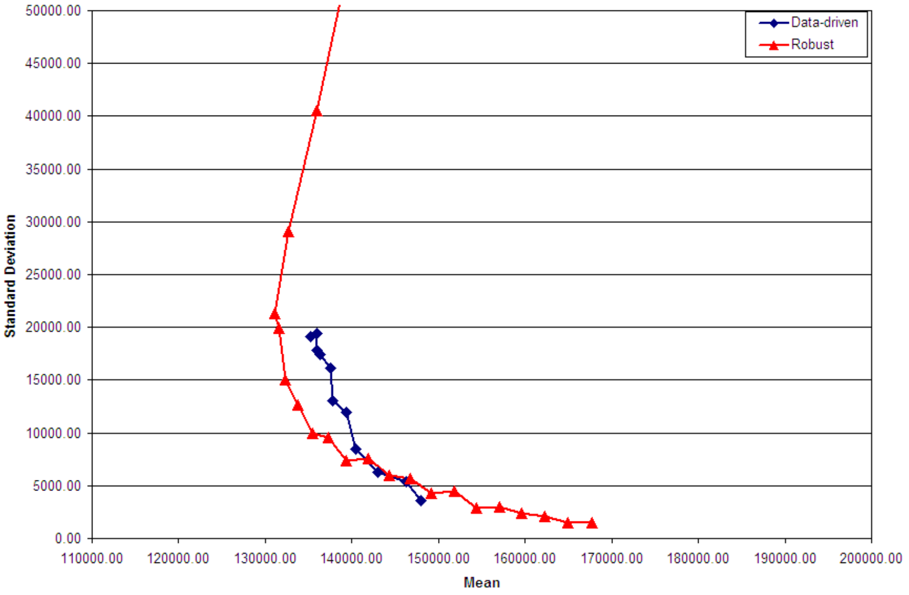

Figure 14: Mean-standard deviation frontiers of two models for the network design $C2$

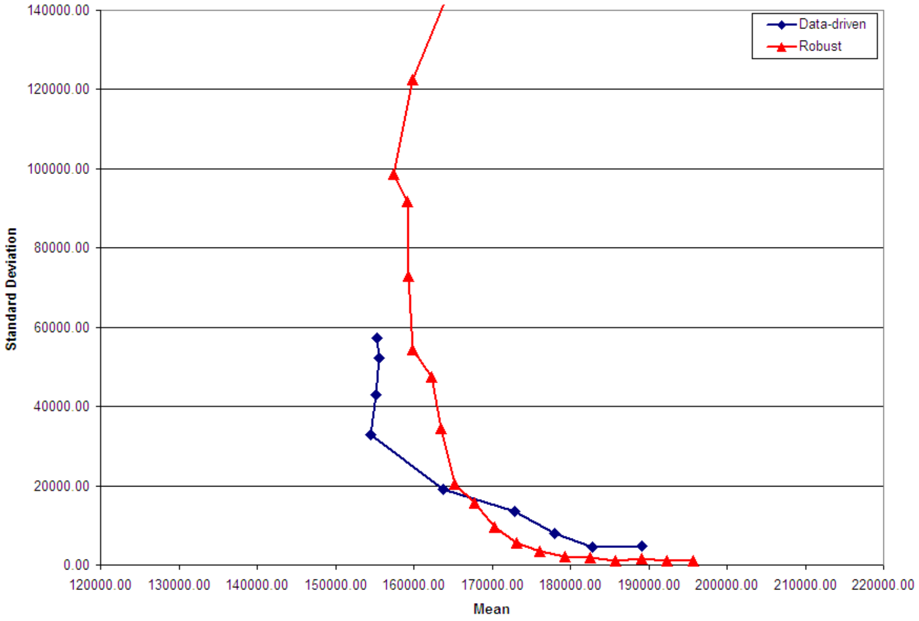

Figure 15: Mean-standard deviation frontiers of two models for the network design $C3$

terms of computational time, the robust model can be solved significantly faster than the data-driven model in all cases. The more complicated network design is, the more significant the time difference is. It shows that the proposed robust model produces better solutions than those obtained from the risk-averse data-driven approach in most experiments with a huge difference in the computational time.

Acknowledgment

We would like to thank the associate editor and two anonymous referees for their comments on the earlier version of this paper. We also want to thank Professor Mandelbaum from the Technion Service Enterprise Engineering Center for providing us with the SEESTAT software whose data are used extensively in this paper.

References

[1] A. Bassamboo and A. Zeevi. On a data-driven method of staffing large call centers, 2007. Preprint.

[2] D. Bertsimas and A. Thiele. A data-driven approach to newsvendor problem, 2006. Preprint.

[3] 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.

[4] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5(2):79–141, 2003.

[5] I. Gurvich and W. Whitt. Fixed-queue-ratio routing in many-server systems, 2006. Preprint.

[6] J. M. Harrison and A. Zeevi. A method for staffing large call centers based on stochastic fluid models. *Manufacturing and Service Operations Management*, 7:20–36, 2005.

[7] H. Levy. *Stochastic Dominance: Investment Decision Making under Uncertainty.* Springer, second edition, 2006.

[8] S. Ross. *Simulation.* Academic Press, second edition, 1997.

[9] V. Trofimov, P. Feigin, A. Mandelbaum, E. Ishay, and E. Nadjharov. DATA-MOCCA: Data MOdel for Call Center Analysis. Volume 1: Model description and introduction to user interface, July 2006.