Can “Very Noisy” Information Go a Long Way?
An Exploratory Analysis of Personalized Scheduling in Service Systems

Seyed Emadi
University of North Carolina at Chapel Hill, Kenan-Flagler business school, Seyed_Emadi@kenan-flagler.unc.edu

Rouba Ibrahim
University College London, School of Management, rouba.ibrahim@ucl.ac.uk

Saravanan Kesavan
University of North Carolina at Chapel Hill, Kenan-Flagler business school, kesavans@kenan-flagler.unc.edu

This version: January 10, 2019

Problem definition. In this work, we focus on the implementation of personalized scheduling policies which exploit noisy individual customer information. In particular, we consider the shortest job first policy with noisy service time predictions.

Academic/Practical Relevance. The extant queueing-theoretic literature on personalized scheduling typically assumes perfect information about customers, while in practice such information tends to be “noisy”. Therefore, it remains unclear to what extent the promise demonstrated in theoretical work can translate into practice. In this paper, we investigate the tradeoff between information availability and operational performance by studying the performance of noisy service-time-based scheduling.

Methodology. We use a combination of three methodologies. Through our empirical study, we investigate how personalized scheduling using service-time predictions from a real call-center performs in practice. Motivated by our empirical results, we use both simulation modeling and queueing theory to deepen our understanding into that operational performance.

Results. We present empirical and theoretical evidence that “very noisy” information can go a long way in personalized service-time-based scheduling. First, we quantify the improvement in predicting service times when exploiting the individual histories of callers, which enables us to implement customer-specific scheduling. Second, we perform detailed simulation studies which quantify the performance of noisy service-time-based scheduling in a variety of queueing contexts. Third, we derive sufficient conditions under which the expected waiting time with noisy service-time-based-scheduling is smaller than for any finite priority-based scheduling policy.

Managerial Implications. Our analysis shows that call center managers can benefit substantially from implementing personalized scheduling disciplines, even when personalized customer information is considerably noisy. We derive insights on the performance of noisy service-time-based scheduling in general service systems. For example, we show that such scheduling performs better in large and congested systems.

Key words: personalized queues; partial information; shortest job first; service times; call centers.
1. Introduction

The rapid proliferation of customer-specific data is making the personalization of products and services increasingly practice-relevant (Rossi et al. 1996, Acquisti and Varian 2005, Forbes 2018, McKinsey 2018). Personalized scheduling where, e.g., service durations or (im)patience times are assumed to be known for each individual customer, holds great promise to improve operational performance compared to classical protocols such as first-come-first-served (FCFS) or last-come-first-served (LCFS) which lack the customer-specific information. For example, an extensive body of queueing-theoretic literature demonstrates that assigning priority to the customer with the shortest service time or with the smallest patience level can yield superior performance compared to blind, size-unaware policies (e.g., early references include Schrage and Miller (1966) and Schrage (1968)). However, this literature typically assumes perfect information about customers, while in practice such information tends to be “noisy”. Therefore, it remains unclear to what extent the promise demonstrated in theoretical work can translate into practice.

A fundamental tradeoff. At the crux of personalized operational decision-making lies a fundamental “tradeoff between information availability and operational performance” (Mandelbaum and Momčilović 2017). Accurate customer information can lead to better operational performance when exploited through personalized scheduling, but such information may be either costly or impossible to obtain. For example, in the context of a call center, estimates of call durations are notoriously imprecise because service times are driven by the randomly varying work content of various customer questions and requests, which are difficult to predict ex-ante (Gans et al. 2010). This raises fundamental questions on the feasibility of operationalizing personalized scheduling in practice.

In this paper, we investigate that tradeoff between information availability and operational performance using data from a bank call center. Specifically, we investigate operational performance with personalized, size-aware, call-routing disciplines that schedule customers according to noisy estimates of their anticipated service demands, i.e., using partial service-time information.

Personalized scheduling: From prediction to operational performance. There are two alternative approaches to personalizing service-time information: (i) a personalized server view, where servers are heterogeneous, e.g., in their processing speeds (Gans et al. 2010, Ibrahim et al. 2016); and (ii) a personalized customer view, where customers are heterogeneous in the specific realizations of their service requirements, i.e., while the underlying distribution of service times is the same across all customers. This is unlike e.g., skill-based-routing where there are multiple customer classes and alternative, class-specific, distributional characteristics (Brown et al. 2005). In this paper, we combine both the personalized server and personalized customer views. In particular, we adopt
an empirical approach to extract information about individual customers and individual servers
based on their histories in the system. For servers, we use agent fixed effects which capture the
impact of agents’ individual abilities on the durations of calls that they handled. For customers,
we use previous waiting times, abandonment decisions, service durations, etc., i.e., entire customer
histories. (Our data set contains unique customer identifiers, and many repeat callers, which allows
tracking each customer’s history in the call center.) Thus, we are able to differentiate both among
servers and among customers to explain more variability in the service times. As such, we obtain
more accurate predictions of future service durations, compared to e.g., differentiating solely among
servers, which has been the predominant view in the literature (Gans et al. 2010, Ibrahim et al.
2016). We investigate how those superior predictions may be used in scheduling customers.

Differentiating both among servers and among customers allows for the implementation of size-
aware scheduling policies, which would not be possible otherwise. For example, with heterogeneous
servers, one could implement policies such as fastest-server-first (Armony 2005). Also, with hetero-
genous customers, one can implement policies such as shortest-job-first, SJF (Wierman 2007b). In
general, such size-aware policies have been shown to yield superior performance compared with blind,
size-unaware policies, e.g., such as FCFS. To test if this theoretical prediction holds with real-world
noisy call-center data, we describe results of simulation experiments which explore the performance
of the SJF policy in our call center using our empirical predictions of service times, derived as above.
We find that using personalized information for both servers and callers can improve the accuracy of
service-time predictions, albeit not to a great extent. For example, we observe a 5% increase in the
$R^2$ of the out-of-sample prediction when we use callers’ historical information. However, the overall
$R^2$ remains low, at 8.5%. This low predictive accuracy is consistent with the prior literature (Gans
et al. 2010, Ibrahim et al. 2016). Even with such noisy service-time predictions ($R^2=8.5\%$), our
simulation results reveal that, for the call center at hand, the SJF policy can lead to between 32%
and 48% decrease in average waiting times and abandonment rates, compared to the FCFS policy.
This shows that the call-center manager can benefit substantially from implementing personalized
scheduling disciplines, even when this personalized information is considerably noisy.

Motivated by that empirical evidence, we devote the rest of the paper to deepening our under-
standing of operational performance under noisy service-time-based scheduling. We do so through:
(i) a counterfactual numerical study in general queueing systems; and (ii) a queueing-theoretic
approach where we derive theoretical results in a parsimonious single-server queueing model. In
our counterfactual study (item i), we vary alternative parameters in the system, such as the arrival
rate, the time-to-abandon distribution, and the number of servers, and explore the performance of
noisy service-time-based scheduling in each case. As such, we glean generalizable insights on that performance that hold broadly across alternative service systems. For example, we find that implementing the “noisy” SJF policy leads to smaller queues and shorter overall waiting times, compared to FCFS, as system congestion increases, or as the system size increases. It is well known that service-time-based scheduling may unfairly favor small jobs (Wierman and Harchol-Balter 2003). To address the issue of unfairness, we also propose heuristic policies that limit the effect of unfairness towards customers while not compromising on the superior performance of noisy service-time-based scheduling. In our theoretical analysis (item ii), we derive sufficient conditions under which implementing the SJF policy using noisy service-time information leads to a smaller expected waiting time than any finite priority-based scheduling policy. A finite priority-based scheduling policy is one where customers are partitioned into a finite number of priority classes based on their service-time estimates. Customers within each class are served according to FCFS. Note that FCFS is a finite priority-based scheduling policy with only one such priority class. Those sufficient conditions pertain to the pair of random variables characterizing the true and estimated service times, and hold for the service-time models of our empirical study.

Main contributions and organization. A main contribution of this paper is the illustration of the following insight: “Very noisy" information can go a long way in personalized scheduling. A key challenge with implementing personalized scheduling in practice is the lack of accurate information about customers. Using call-center data, we show that even though there exists a trade-off between information availability and operational performance, it is still possible to benefit by implementing size-aware scheduling: It can lead to shorter queues and lower overall waiting times, even when the information is considerably noisy. Another contribution of this paper lies in the prediction of service times for individual customers. Call durations are notoriously difficult to predict, as they are the result of “many sources of systematic variation” (Gans et al. 2003). Prior literature has used agent-specific information as a useful predictor of service times (Gans et al. 2010, Ibrahim et al. 2016). In this work, we are the first, to the best of our knowledge, to use customer contact history information for service-time prediction. We show that customer history, when available, contains incremental information which is useful to predict service times.

On the theoretical front, we derive the expected waiting time under the SJF policy with noisy service-time information in a single-server queue. We also derive sufficient conditions, on the pair of random variables representing the actual and estimated service times, for that expected waiting time to be smaller than for any finite priority-based scheduling policy, e.g., including FCFS (where there is only one such priority class). Despite a long literature in size-aware scheduling, almost all
analytical results are based on assumptions of perfect information on service times. To the best of our knowledge, the only exception is Wierman and Nuyens (2008) which also considers a single-server model. However, that paper does not consider non-preemptive policies, such as SJF, as we do here. Non-preemptive policies are especially practice-relevant in call centers, since interrupting a phone conversation to answer another one would be detrimental for customer service.

To the best of our knowledge, this work is one of the first attempts at marrying theory with data to provide a better understanding of the practical performance of personalized service-time-based scheduling, in particular the SJF policy. We believe that doing so is important, particularly because despite having been studied extensively in the theoretical literature, the effectiveness and implementation challenges of the SJF policy have not been studied using real-world data sets. Combining data analysis, simulation modelling, and queueing theory, as we do in this paper, is in line with the celebrated “Theompirical Approach” to managing service operations (Mandelbaum 2017). We expect this approach to become increasingly prevalent in the future, particularly with the proliferation of detailed customer-specific data sets.

The remainder of the paper is structured as follows: In Section 2, we present the literature review. In Section 3, we describe the data set and lay out the service-time prediction models and their performance. In Section 4, we describe results of simulation studies that investigate the performance of the SJF policy under different scenarios and with noisy predictions. In Section 5, we present theoretical results in a non-preemptive single server setting. And finally, in Section 6, we present concluding remarks. We relegate technical proofs and additional numerical results to the appendices, and present supporting numerical evidence in an online supplement to the paper.

2. Literature Review

The personalization of products and services based on customer/task information has been an important business strategy in different contexts, e.g., as in the customization of marketing efforts (Rossi et al. (1996) and Reinartz and Kumar (2003)), the recommendation of products and services (Oard et al. (1998) and Rendle et al. (2009)), the prediction of customer churn (Wei and Chiu (2002) and Huang et al. (2012)), and judiciary case hearing policy (Bray et al. (2016)). In the queueing literature, the focus has been on the personalization of scheduling policies based on customer information. This literature can be divided into two streams. The first stream studies scheduling policies based on customer patience-level information, and the second stream focuses on scheduling policies based on customer service-time information.

The stream of literature on scheduling using patience-level information is extensive, e.g., studying the least-patient-first or earliest-deadline-first policy (Wein 1991, Stoyenko and Georgiadis 1992,
Moyal 2013). Almost all works in this stream of literature assume that customers’ patience level is perfectly known, however some notable exceptions are Mandelbaum and Momčilović (2017), Emadi and Swaminathan (2018), and Hathaway et al. (2018). Mandelbaum and Momčilović (2017) show that scheduling the least patient customer first can lead to substantial reduction in waiting times and abandonment proportions, and use a fluid approximation to study performance when patience-level information is noisy. Emadi and Swaminathan (2018) show that customers learn about the waiting-time distribution from their past waiting experiences, and provide a Bayesian learning model to estimate customers’ patience level based on their contact history. Hathaway et al. (2018) study scheduling policies based on callers’ patience level and their likelihood of redialling where customer contact history is being used to estimate their abandonment and redial probabilities.

The stream of works on scheduling using customer service times starts with Schrage and Miller (1966) which studies the shortest remaining processing time (SRPT) policy for a single server $M/G/1$ system. The SRPT policy allows preempting the customer currently in service, whereas the shortest-job-first (SJF) policy, which we consider in this paper, does not allow for such preemptions. Schrage (1968) proves the optimality of the SRPT policy. Later works include Wierman (2007b) which studies different heuristic approaches to implement the SRPT policy in practice, as well as Down et al. (2009) and Gromoll et al. (2011) which consider a heavy-traffic framework of analysis. Almost all works on the SJF or SRPT policies consider a single-server model. The only exception, which studies multiserver systems, is Grosof et al. (2018). Grosof et al. (2018) assumes perfect knowledge of service times, and studies a preemptive SRPT scheduling policy. There, the authors show that this policy has an asymptotically optimal mean response time in heavy traffic. Despite the remarkable improvement in performance achieved by the SRPT and SJF policies, this class of policies suffers from a fairness issue (Bansal and Harchol-Balter 2001), Wierman and Harchol-Balter (2003) and references therein). We address this issue later on in our paper as well.

Our work is different from the extant literature on service-time-based scheduling in that the extant literature typically assumes perfect knowledge about service times, whereas we consider a practical setting in which information about service times is noisy. To the best of our knowledge, there are very few papers which consider scheduling under noisy service-time information. Mailach (2017), Dell’Amico et al. (2014), Lu et al. (2004) describe results of numerical studies quantifying the impact of noisy service-time information in scheduling, but they do not consider systems which mimic the operations of real-life service systems (e.g., multiserver queueing systems with many servers, time-varying arrivals, and customer abandonment) as we do here, nor do they address the issue of predicting those service times. Predicting customer service times is a notoriously cumbersome
task with low predictive accuracy as indicated by Gans et al. (2010) and Ibrahim et al. (2016). The major difference between our work and these two papers is that, to the best of our knowledge, we are the first to track customer histories and use both past customer information and past agent information to predict future service times. On the theoretical front, Wierman and Nuyens (2008) is the only paper that we are aware of which presents analytical results on the performance of SRPT with noisy information in a single-server setting. However, they do not present theoretical results on performance with non-preemptive policies, such as SJF, as we do here.

3. Data Description and Analysis

Our data set contains individual call-level data from a bank call center. The time span of the data set is from April of 2007 to June of 2009. The call center provides 6 types of services ranging from checking/saving account services to loans to 4 classes of customers: high priority, medium priority, low priority and no priority. We focus on the medium priority group, which is the largest priority group, and service type 1, which contains more than 80% of the observations. This portion of data contains 3,874,396 calls made by 326,270 customers. The average waiting time and abandonment rate of customers in this portion of data are 53.33 seconds and 4.49%, respectively. The average service time of callers is 225.61 seconds. For a robustness check, we have repeated our analyses in the subsequent sections on other priority groups and service types and have not seen a significant change in the insights and results of our analyses.

For each call in the data set, we can observe the time and date of arrival of that call, the amount of time spent in the queue, the outcome of the call (abandoning vs. talking to an agent) and, if the caller did not abandon and eventually received service, the service time and the ID of the handling agent. We can also observe the identifier (ID) of the customer who made the call. Observing customer ID’s enables us to track customers and the history of their past contacts. On average, callers contact the call center 11.87 times during the time span of the data set. The average time between two consecutive contacts of callers is 30.85 days. Furthermore, on average on a given day, only 8.42% of arriving calls are made by customers with no history. In other words, more than 91% of the arriving calls on a given day are made by customers who contacted before. This shows that the call center has contact history information for the majority of arriving customers, which can be leveraged in scheduling policy decisions. On average, around 4800 callers from our focus group (medium priority callers who requested service type 1) contact the call center daily.

Figure 1 shows the average number of arrivals and average service times for callers in our sample at different times of the day. As can be seen in Figure 1, most of the callers from our focus group contact
the call center between 8am and 1pm. In other words, the rush hours for our sample correspond to the time slot between 8am and 1pm. Moreover, the average service time ranges between 170 seconds and 240 seconds; however, it does not change significantly during rush hours.

The number of agents who worked in the call center during the time span of the data set is 1,835. However, the average number of agents working on a given day is 400. On average, each agent in our data handles 2,017 calls from our focus group. There is significant variability across agents in terms of their average service time; see Figure 2 for a histogram of these average service times. As can be seen in Figure 2, most of the agents provide service in around 200 seconds. However, there is a significant portion of agents who provide service either much faster or much slower.

3.1. Service Time Prediction

In this section, we use the observable variables in the data to predict customer service times using a series of regression models. Using different sets of variables and a variety of models, we show that the out-of-sample $R^2$ of service-time prediction does not exceed 10%. In other words, none of our models can achieve accurate predictions of service times and, at best, our predictions remain quite noisy. This is not surprising given that service durations in call centers are notoriously difficult to predict (Gans et al. 2003). Indeed, our overreaching objective is to illustrate that even such noisy service-time estimates remain useful for service-time-based scheduling purposes.

We first explain our notation for the variables in the data. Then, we lay out the set of variables used for service-time prediction, and the different regression models considered, utilizing these variables. Next, we explain our strategy to assess the predictive performance of different models. Finally, we compare the different models in terms of that predictive performance.

**Preliminaries.** Denote by $N$ the number of callers and by $i \in \{1, \ldots, N\}$ the index of callers in the data set. Let $n_i$ denote the number of times caller $i$ contacts the call center in the data set.
Similarly, let $M$ denote the number of agents and let $j \in \{1, \ldots, M\}$ denote the index of agents. Moreover, let $c_i$ denote the fixed effect for customer $i$, and let $[h_{in}]_{24 \times 1}$, $[dow_{in}]_{7 \times 1}$, $[m_{in}]_{12 \times 1}$ and $[a_{in}]_{M \times 1}$ denote the vectors of indicator variables for hour of the day, day of the week, month, and the agent handling the call for caller $i$’s $n^{th}$ contact.

Denote by $s_{in}$, $w_{in}$ and $o_{in}$ the service time, the waiting time, and the outcome of the call for caller $i$’s $n^{th}$ contact. Note that service time $s_{in}$ only applies to callers who receive service. The outcome variable $o_{in}$ is equal to 1 if the caller abandons, and is equal to 0 if the caller receives service, i.e., does not abandon. We also assume that $x_{i(-k)}$, where $x \in \{s, w, o\}$, denotes the variable $x$ lagged by $k$ contacts. For example, $s_{i(-1)}$ denotes the service time lagged by 1 contact i.e., the previous service time of caller $i$. Furthermore, let $H_{in} = \left[ [s_{ik}]_{(k=1\ldots n-1)}, [w_{ik}]_{(k=1\ldots n-1)}, [o_{ik}]_{(k=1\ldots n-1)} \right]$ denote the vector of caller $i$’s past contact history before her $n^{th}$ contact. Similarly, let $H_{i(-q)} = \left[ [s_{i(-k)}]_{(k=1\ldots q)}, [w_{i(-k)}]_{(k=1\ldots q)}, [o_{i(-k)}]_{(k=1\ldots q)} \right]$ denote the vector of caller $i$’s contact history variables lagged by 1 to $q$ contacts. We also let $av(H_{in}) = [av(s_{in}), av(w_{in}), av(o_{in})]$ denote the vector of averages of caller $i$’s contact history variables before contact $n$, i.e., from contact 1 to contact $n-1$. In addition, given that callers may make a follow-up call after talking to an agent, i.e., they may be a retrial, which may affect the duration of the service, we define the retrial indicator variable $r_{in}$ for caller $i$’s $n^{th}$ contact. The variable $r_{in}$ is equal to 1 if the caller talks to an agent in her $(n-1)^{th}$ contact, and calls back again in less than 24 hours, i.e., we assume that if a caller contacts again within 24 hours, then the caller wants to talk about the same issue and the call is a follow-up call.

### 3.1.1. Prediction Models.

In this section, we first lay out the observable covariates in the data that can be used for service-time prediction, and explore their explanatory power. Then, we describe different regression models to predict service times.

We divide the observable covariates in our data to the following sets:

- **Time Fixed effect variables (TF):** $[h_{in}]_{24 \times 1}$, $[dow_{in}]_{7 \times 1}$, and $[m_{in}]_{12 \times 1}$.
- **Agent Fixed effect variables (AF):** $[a_{in}]_{M \times 1}$.
- **Detailed History variables (DH):** past contact history $H_{in} = \left[ [s_{ik}]_{(k=1\ldots n-1)}, [w_{ik}]_{(k=1\ldots n-1)}, [o_{ik}]_{(k=1\ldots n-1)} \right]$.
- **Average History variables (AH):** average contact history $av(H_{in}) = [av(s_{in}), av(w_{in}), av(o_{in})]$.
- **Retrial variables (RT):** $r_{in}$ and $r_{in} \times s_{i(-1)}$. We consider the retrial indicator variable and its interaction with the service time of the previous contact to explore the impact of retrials on service times.
- **Customer Fixed effect variables (CF):** $c_i$. 
The time and agent fixed effect variables, TF and AF, have been used in the extant literature (Ibrahim et al. (2016) and Gans et al. (2010)) to predict service times in call centers. To the best of our knowledge, this work is the first one that explores the role of customer history, DH, AH, RT, and customer fixed effect, CF, covariates in predicting service times.

Figure 3 shows the histogram of service times in our data set and a lognormal distribution fitted to the service-time distribution, which is depicted by a solid red line. The fitted lognormal distribution is Lognormal(5.05, 0.82) where $\mu = 5.05$ and $\sigma^2 = 0.82$. As can be seen in Figure 3, the distribution of service times has a remarkable fit to the lognormal distribution. This observation is consistent with the findings of Brown et al. (2005). Consequently, in all regression models in this paper, the dependent variable is the natural logarithm of the service time $\log(s_m)$.

![Figure 3](image)

**Figure 3** The histogram of customers’ service times and the fitted lognormal distribution.

To explore the explanatory power of the aforementioned covariates, we run the following set of regressions on our data set.

**Regression on TF covariates:** We first run a regression on time fixed effect covariates. The $R^2$ of this regression is 0.23% and is statistically significant at the 99% confidence level, i.e., it yields a p-value $p < 0.01$. Most of the time fixed effect variables for this regression are significant at $p < 0.01$. For brevity, we do not report the coefficients of this regression.

**Regression on TF+AF covariates:** Running a regression on the time and agent fixed effect variables leads to an $R^2$ equal to 5.30%, which is statistically significant at $p < 0.01$. Most of the fixed effects are statistically significant at $p < 0.01$. For brevity, we do not report the coefficients of this regression. The increase in $R^2$ from 0.23% to 5.30% shows that agents are heterogeneous in terms of their service speeds, and taking their fixed effects into account would increase the explanatory
power of the regression model significantly.

**Regression on TF+AF+RT+DH:** Given that callers contact the call center different numbers of times, which leads to different history lengths, callers’ past contact history vectors $H_{i,n}$ have different lengths. Consequently, we are dealing with an unbalanced sample and cannot run a single regression for all contacts in the data. Hence, we run a series of regressions on time fixed effects, agent fixed effects, retrial, and detailed history covariates beyond callers’ different numbers of contacts, where the regressions vary in terms of callers’ history lengths. For example, for the regression on callers’ contacts beyond their third contact, we focus on a sub-sample of the data where callers contacted at least three times, and consider the set of covariates: time fixed effects, agent fixed effects, retrial variables, and callers’ history variables lagged by 1 to 3 denoted by $H_{i(-3)}$. Note that for contacts beyond 3, at least 3 lagged history variables are available. For the regression on TF+AF+RT+DH covariates beyond callers’ first contacts, the $R^2$ is 8.02%. If we run this regression on covariates beyond callers’ 10th contact, then the $R^2$ increases to 12.65%. Running the regression for an even higher number of contacts, which amounts to a longer history for the caller, would increase the $R^2$; however, we found that this $R^2$ does not exceed 13% even for the longest history in the data (69 contacts). We also considered an alternative model-estimation strategy where instead of focusing on the sample of callers who have contacted at least $n$ times i.e., $n+1$, $n+2$, times etc., we ran the same regression on TF+AF+RT+DH variables for the sample of callers who have contacted exactly $n$ times. We obtained roughly the same $R^2$ under both estimation strategies.

To get a sense of the impact of callers’ history, we next explain the regression results on TF+AF+RT+DH covariates beyond callers’ third contacts. The insights do not change for other contact history lengths. Table 1 shows that the $R^2$ of the regression on TF+AF+RT+DH covariates beyond callers’ third contacts is 10.29%. Comparing this value with the $R^2$ of the regression on agent and time fixed effects (5.30%) quantifies the amount by which callers’ past history information can increase the explanatory power of the service-time prediction.

As can be seen in Table 1, the impact of all contact history variables on callers’ service times is statistically significant. Consequently, callers with different history information will have statistically different service times. Moreover, the coefficients for the average past service times $s_{in}$ and lagged service times are positive, which indicates that callers with longer service times in the past will have longer service times in the future. This observation shows that callers are intrinsically different in terms of the duration of time needed for the completion of their requested services, and these intrinsic differences can be captured using their contact history information.
Moreover, as can be seen in Table 1, the coefficient for the retrial indicator variable \( r_{in} \) is negative and the coefficient for its interaction with the past service time \( r_{in} \times s_{i(-1)} \) is positive. These observations show that follow-up calls take a shorter amount of time; however, their duration is positively correlated with that of the last call.

Next, to evaluate the predictive power of callers’ average history versus their lagged history variables, we replace customers’ detailed history with their average history and run a regression on TF+AF+RT+AH covariates beyond callers’ first contact. Note that callers do not have any history in their first contact. Consequently, first-contact observations cannot be included in this regression. Table 2 shows the regression results on TF+AF+RT+AH covariates.

As can be seen in Table 2, all average history variables are statistically significant. Also, the \( R^2 \) of the model with TF+AF+AH+RT covariates is 10.66%. Recall that the \( R^2 \) that can be achieved using TF+AF+AH+RT covariates, which includes callers’ detailed history information, does not exceed 13% even for customers with a very long history. This shows that customers’ average history captures most of the information contained in callers’ detailed history.

Regression on TF+AF+RT+DH+CF: We run a series of regressions on time fixed effects, agent fixed effects, retrial variables, detailed history covariates and customer fixed effects, where the regressions vary in terms of callers’ history lengths. The \( R^2 \) of these regressions ranges between 24.20% to 24.50%. Table 3 shows the regression results on TF+AF+RT+DH+CF covariates beyond callers’ third contacts. The insights of our analyses do not change for other contact history lengths.

As can be seen in Table 3, the \( R^2 \) of the regression that includes customer fixed effects on top of time fixed effects, agent fixed effects and detailed history variables is 24.35%. This shows that including customer fixed effects almost doubles explanatory power of the regression from 10.29% to 24.35%.
**Denotes statistically significant at $p < 0.05$. Time, agent and customer effects are omitted for brevity.

<table>
<thead>
<tr>
<th>Dep. variables</th>
<th>Coefficient</th>
<th>St. Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>5.1130**</td>
<td>0.1988</td>
</tr>
<tr>
<td>$r_{in}$</td>
<td>-0.0252**</td>
<td>0.0018</td>
</tr>
<tr>
<td>$r_{in} \times s_{i(-1)}$</td>
<td>0.0002**</td>
<td>5.74E-06</td>
</tr>
<tr>
<td>$s_{i(-1)}$</td>
<td>-3.33E-06</td>
<td>2.59E-06</td>
</tr>
<tr>
<td>$w_{i(-1)}$</td>
<td>4.89E-05**</td>
<td>5.27E-06</td>
</tr>
<tr>
<td>$o_{i(-1)}$</td>
<td>0.0380**</td>
<td>0.0025</td>
</tr>
<tr>
<td>$s_{i(-2)}$</td>
<td>-4.10E-05**</td>
<td>2.39E-06</td>
</tr>
<tr>
<td>$w_{i(-2)}$</td>
<td>2.73E-05**</td>
<td>5.19E-06</td>
</tr>
<tr>
<td>$o_{i(-2)}$</td>
<td>-0.0056**</td>
<td>0.0024</td>
</tr>
<tr>
<td>$s_{i(-3)}$</td>
<td>-0.0001**</td>
<td>2.42E-06</td>
</tr>
<tr>
<td>$w_{i(-3)}$</td>
<td>1.15E-05**</td>
<td>5.14E-06</td>
</tr>
<tr>
<td>$o_{i(-3)}$</td>
<td>-0.0130**</td>
<td>0.0024</td>
</tr>
</tbody>
</table>

Table 3 The regression results for TF+AF+RT+DH+CF covariates beyond callers’ third contacts. $R^2 = 24.35\%$.

Our regression results show that the different set of covariates (time fixed effects, agent fixed effects, history variables and customer fixed effects) have explanatory power with the $R^2$ ranging in values between 0.23\% to 24.50\%. However, for the purposes of prediction, we need to examine how different regression models with these set of covariates perform in out-of-sample tests. We assess the out-of-sample predictive power of the following regression models in the next section. For callers with no history, only time and agent fixed effects will be used.

**Model 1** - Using TF (time fixed effect) covariates for prediction.

**Model 2** - Using TF+AF (time and agent fixed effects) covariates for prediction.

**Model 3** - Using TF+AF+RT+DH (time fixed effects, agent fixed effects, retrial and detailed history) covariates for prediction.

**Model 3’** - Using TF+AF+RT+AH (time fixed effects, agent fixed effects, retrial and average history) covariates for prediction. Model 3’ is a more parsimonious version of Model 3 after replacing the detailed history variables $H_{in}$ with the average history variables $av(H_{in})$.

**Model 4** - Using TF+AF+RT+DH+CT (time fixed effects, agent fixed effects, retrial, detailed history and customer fixed effect) covariates for prediction.

3.1.2. **Assessment of Predictive Performance.** To assess the predictive performance of the regression models, we use a series of out-of-sample tests. These out-of-sample tests are designed to imitate how prediction models are actually used in the call center setting. To be more specific, in call centers, historical data is used to train a model and then the trained model is used to make prediction for a period of time. Then, the model is rolled and updated at the end of the prediction period to be used in the next prediction period. Hence, we do not divide our data to a training set and a validation set randomly but we do it chronologically.

As we mentioned in Section 3, our data set spans from April of 2007 to June of 2009. We consider each month in 2009 as a validation set and all observations before that month as the
training set. To be more specific, we consider the following 6 out-of-sample tests: (For \( \text{Month} \in \{\text{Jan. 2009, Feb. 2009,\ldots, June 2009}\} \), Training set: All observations from April 2007 to \( \text{Month}-1 \), Validation set: All observations in \( \text{Month} \)).

We estimate the parameters of the models on the training set and predict the logarithm of the service time for the observations in the validation set. When using Model 3 or Model 4 (with callers’ detailed histories) to predict service times in the validation set, we must account for the fact that different callers have different history lengths in the data. To do so, we fit a different regression model depending on each specific caller’s history length. For example, to predict a caller’s service time for her \( n^{th} \) contact in the observation set, we consider versions of Models 3 and 4 where the independent history variables consist of the previous \( n - 1 \) contacts, and the dependent variable is the service time on the \( n^{th} \) contact. To fit this model to data in training set, we focus on the sub-sample of callers who have contacted at least \( n \) times in that set.

We find the \( R^2 \) and root-mean-square error, RMSE, for the predicted values in the validation set for the 6 out-of-sample tests and consider their averages across the 6 tests as the performance metrics for the prediction models. Suppose that there are \( K \) observations in the validation set, where \( \log s_l \) and \( \log \hat{s}_l; \ l = 1..K \), denote the actual and predicted logarithm of service times in the validation set. The \( R^2 \) and RMSE are given by:

\[
R^2 = 1 - \frac{\sum_{l=1}^{K}(\log s_l - \log \hat{s}_l)^2}{\sum_{l=1}^{K}(\log s_l - \text{mean}(\log s))^2}, \quad \text{where} \quad \text{mean}(\log s) = \frac{\sum_{l=1}^{K} \log s_l}{K},
\]

\[
\text{RMSE} = \sqrt{\frac{\sum_{l=1}^{K}(\log s_l - \log \hat{s}_l)^2}{K}}.
\]

3.1.3. Results and Comparisons. Table 4 shows the averages of \( R^2 \) and RMSE across the 6 out-of-sample tests for the predicted logarithm of service times for Model 1 to Model 4, and Model 3’ for callers in the validation set. For the RMSEs (and \( R^2 \)s), we ran two-sample t-tests on the RMSEs (and \( R^2 \)s) of the 6 out-of-sample tests of all model pairs to see if the differences between the performance metric averages of the models are statistically significant at the 95% confidence level, i.e., if the p-values are less than 0.05. Our t-test results show that we can divide the models into the following clusters in terms of their performance metrics: [Model 1], [Model 2, Model 4], and

<table>
<thead>
<tr>
<th>Measure</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 3’</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R^2 )</td>
<td>-2.822%</td>
<td>3.323%</td>
<td>8.819%</td>
<td>8.462%</td>
<td>3.247%</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.873</td>
<td>0.846</td>
<td>0.820</td>
<td>0.823</td>
<td>0.846</td>
</tr>
</tbody>
</table>

Table 4 Predictive performance metrics for the logarithms of the service times.
Models in the same clusters are statistically the same and models in different clusters are statistically different in terms of their performance metrics.

The comparison between the performance measures of Model 2 and Model 1 shows that using agent fixed effect increases the $R^2$ by 6% and decreases the RMSE by 0.03. However, none of these models (Model 1 and 2) differentiate between individual callers. Thus, relying on predictions using these models would not allow to implement customer-personalized scheduling policies such as the SJF policy. In contrast, service-time predictions based on Model 3, Model 3′, and Model 4 can be used for the implementation of the SJF policy as these models can differentiate between callers and their individual service times by exploiting their histories and customer fixed effect variables.

As can be seen in Table 4, Model 3′ which utilizes customer average history information has a prediction performance close to that of Model 3 which utilizes their detailed history information. In particular, the difference between the RMSE of these models is in order of 0.001, and the difference between their $R^2$ is less than 0.04%, and both differences are not statistically significantly different than 0. This shows that customer average history variables are sufficient statistics to capture almost all information in customers’ contact history relevant to predicting their service times. Given that Model 3′ is more parsimonious, we prefer it to Model 3. The comparison between Model 3′ and Model 2 shows that adding customers’ average history information increases the $R^2$ by 5% and decreases the RMSE by 0.02. However, adding customer fixed effects in Model 4 leads to over-fitting and a deterioration of predictive performance. Recall from Table 3 that the in-sample $R^2$ of Model 4 can reach 24%. However, due to the over-fitting issue, its out-of-sample $R^2$ is less than 4%.

Overall, based on our out-of-sample test results, and considering the parsimony principle and predictive performance of the candidate models, we choose Model 3′ with time fixed effects, agent fixed effects and customer average history information to predict customers’ service times in our subsequent analyses. For robustness, we repeated the analysis in this section for the high and low priority groups. Recall that we focused on the medium priority group because it has the highest number of observations. Consistent with the medium priority group, Model 3′ provides the best combination of parsimony and predictive performance for the high and low priority groups. Table 5 shows the comparison between the predictive performance of Model 3′ for the high, medium, and low priority groups.

As can be seen in Table 5, Model 3′ performs the best for the medium priority group, which has the highest number of observations and the highest contact frequency. Moreover, Model 3′ has a better performance for caller groups with longer histories (which manifests itself as higher number of observations and higher contact frequencies).
<table>
<thead>
<tr>
<th>Model 3′</th>
<th>Number of callers</th>
<th>Average number of contacts</th>
<th>$R^2$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>High priority</td>
<td>865,861</td>
<td>6.51</td>
<td>5.42%</td>
<td>0.888</td>
</tr>
<tr>
<td>Medium priority</td>
<td>3,874,396</td>
<td>11.87</td>
<td>8.46%</td>
<td>0.823</td>
</tr>
<tr>
<td>Low priority</td>
<td>2,249,173</td>
<td>7.61</td>
<td>7.50%</td>
<td>1.049</td>
</tr>
</tbody>
</table>

Table 5  Predictive performances of Model 3′ for the high, medium and low priority groups.

In addition, in Appendix A, we compare the predictive performance of our chosen regression model (Model 3′) with those of two machine learning models and show that the improvement in the out-of-sample $R^2$ and RMSE if we use the machine learning models will be less than 1% and 0.1, respectively. This shows that Model 3′ not only is a simple and parsimonious model, but it can also provide a predictive performance that rivals that of complex machine learning models.

4. Simulation Study

In the previous section, we illustrated the improvement in goodness-of-fit to data, both in-sample and out-of-sample, which results from incorporating both customer and agent covariates in modelling service times. We now discuss the results of supporting simulation studies. Our objective here is twofold: (i) In §4.1, we gain insight into the performance of noisy service-time-based scheduling in the call center under consideration by using service-time predictions taken from our empirical study; and (ii) in §4.2, we formulate general managerial takeaways on the performance of noisy service-time-based scheduling through a detailed counterfactual study.

4.1. Call Center Case Study

Description of the simulation experiments. We are interested in quantifying operational performance when customers are scheduled according to the noisy service-time estimates of our empirical study in §3. To do so, we select system parameters to mimic performance in the call center under consideration. We restrict attention to the medium-priority group that we have analyzed in §3.1. In reality, the number of agents serving those calls varies with time. Moreover, our call center serves multiple classes of customers, and the medium priority group is served by agents who may be concurrently serving customers from other service groups as well. Since we do not observe the routing policy in our data, and cannot determine how agents are allocated to different customer classes, we assume for simplicity that medium-priority callers are served by a dedicated group of 100 agents. We pick those 100 agents as the top 100 agents who have answered the most calls in the data. In particular, an agent in this cohort has served 8,006 calls on average. The 100 agents in that cohort are heterogeneous in their processing speeds. Call durations depend on both the servicing agent and the served customer.
We simulate the call center in a typical day of operation, between the hours of 8AM and 7PM. We assume a nonhomogeneous Poisson arrival process to the call center. We let the arrival rate be piecewise constant, over successive one-hour intervals. We assume that the system reaches stationarity in each one-hour interval. We calculate average performance measures in the call center, averaged in the entire data set over corresponding one-hour intervals, and select arrival rates in successive one-hour intervals to match those average performance measures (assuming FCFS routing, and 100 dedicated servers serving our callers). In particular, we focus on the waiting time until service, the abandonment proportion, and the overall waiting time until either abandonment or service. We obtain the arrival rates in the first subplot of Figure 4. The arrival rate in the simulation illustrated in Figure 4 is different from the actual arrival pattern in the data (see Figure 1). The reason for this mismatch is that despite the change in the staffing level in the actual call center, in our simulation we do not change the number of agents and vary the arrival rates to match waiting times and abandonment rates instead. Using those arrival rates, we generate arrivals uniformly at random among 100,000 repeat callers (at least 2 calls) to the call center. For these customers, we assume the average customer histories that we observe in the data.

As per our service-time models in §3, we assume that service times are lognormally distributed. To simulate service times, we fit Model 3', which we formulated in §3.1.1, to data. We exclude from this model the retrial component as well as time-of-day effects, because these have a minor impact on both in-sample and out-of-sample goodness of fit. To generate a service time, we add the customer historical component (product of $av(s_{im})$ in Table 2 with its corresponding coefficient) to the agent fixed effect, depending on the identity of the servicing agent. We also add a normally-distributed random variate with standard deviation as estimated from data.

For the routing policy, two types of decisions have to be made. One one hand, if there are several agents who are idle upon arrival of a customer, then we pick the fastest server to serve the newly arriving customer. On the other hand, if there are several customers waiting in line when an agent becomes free, then we pick the shortest service time among those customers waiting in line.

We assume that callers are impatient, and let the time-to-abandon follow an exponential distribution with an average duration of 1,261 seconds, which is the average value that was observed in the data. Our simulation results are based on 10 independent replications of length 12 hours each (this is the duration of the work day that we are simulating). We start each replication with an empty system, and collect performance measures in each one-hour interval. We exclude an initial transient period of length 1,000 arrivals from the beginning of the simulation to reach steady-state conditions. We include detailed tables with all performance measures in the online supplement to this main paper, and focus here on only two measures, as we explain next.
Results. In the second and third subplots of Figure 4, we report point estimates for the average time to abandon or to be served, and the probability to abandon. We let $Q$ denote the steady-state queue length, and $W$ denote the steady-state waiting time until either abandonment or service. We compare performance under the shortest-job-first policy with noisy service time estimates (SJFE) to the standard FCFS policy. Figure 4 illustrates that SJFE consistently outperforms FCFS for the performance measures considered. For example, we see that the mean waiting time is, on average, 40% longer under FCFS than under SJFE (it ranges from a minimum of 32% to a maximum of 48%). We also obtain very similar results for the improvement in estimates of the probability of abandonment. Thus, we conclude that implementing SJFE, based on noisy service-time estimates, can lead to a considerable reduction in the overall congestion level of the call center, despite the fact that the predictive accuracy of our service-time models is admittedly quite poor.

Motivated by these positive results for the performance of SJFE, we describe, in what follows, results from a counterfactual study where we vary alternative parameters in the system. Our objective is to deduce key managerial insights that hold broadly across various systems. In this counterfactual study, we also investigate the key aspect of fairness of the SJF policy which has been previously addressed in the literature in a single-server queuing setting (Wierman 2007a). Here, we describe results quantifying the unfairness of SJF in a multiserver queuing setting with abandonment, and propose heuristic policies that achieve a fairer performance.

![Figure 4] Summary of simulations results for a typical day, from 8AM to 7PM, in the call center.

4.2. Counterfactual Study

Description of the simulation experiments. We simulate the $M/LN/N+M$ queueing system, with a homogeneous Poisson arrival process with arrival rate $\lambda$, $N$ identical servers, lognormal service times, and exponential times to abandon with mean equal to 1. As in Example 4 of Mandelbaum and Momčilović (2017), we capture the partial information in the estimates of service times by
considering correlated lognormal random variables. In particular, we let \((S, \hat{S})\) denote a pair of random variables characterizing the true and estimated service times. We assume that \((S, \hat{S})\) has the same distribution as \((e^Z, e^{\hat{Z}})\), where \((Z, \hat{Z})\) is bivariate normal with equal means and equal standard deviations. We assume that \(E[S] = E[\hat{S}] = 1\) and \(\text{Var}[S] = \text{Var}[\hat{S}] = 1\). Our assumption on the mean service time is done without loss of generality, and amounts to measuring time in units of mean service times. We quantify the dependence between \(S\) and \(\hat{S}\) through the correlation between \(Z\) and \(\hat{Z}\): The higher the correlation, the better the information contained in \(\hat{S}\) in predicting the true service times. Here, we only consider nonnegative values for the correlation. We consider negative values as well in the supplement to the main paper. For each arriving customer \(i\), we generate a pair of random variables \((S_i, \hat{S}_i)\): We let the actual service time of the customer be equal to \(S_i\), and carry out scheduling decisions according to the noisy service-time estimate \(\hat{S}_i\). Our simulation results are based on 10 independent replications of length 2,000,000 arrivals each. In each replication, we discard an initial transient period of length 10,000 arrivals to reach steady-state conditions. In what follows, we describe simulation results based on varying arrival rates, the system size, the time-to-abandon distribution, and the mean time to abandon. In each set of simulations, we compare SJF, SJFE, and FCFS. For SJF, we carry out scheduling decisions assuming perfect knowledge of service times, as benchmark. We focus here on the expected steady-state queue length, \(E[Q]\), but include more detailed results including alternative performance measures in the supplement.

**Congestion in the system.** We begin by investigating the effect of congestion in the system. To vary the system’s congestion level, we keep the number of servers held fixed and equal to 100 servers. Then, we vary the arrival rate \(\lambda\) to change the traffic intensity in the system. In particular, we let \(\lambda\) range from 100 to 160, in increments of 10. The system is moderately loaded for \(\lambda = 100\), and extremely congested for \(\lambda = 160\). The traffic intensity \(\rho \equiv \lambda E[S]/N = \lambda/N\) since \(E[S] = 1\). We also consider alternative values of the correlation between actual and estimated service times.

In order to understand the impact of congestion on the performance of SJFE, we first compare the performances of SJF (without errors) and FCFS as a function of \(\lambda\) (first subplot of Figure 5). Then, we compare SJFE and SJF as a function of \(\lambda\) (second subplot of Figure 5). Finally, we compare SJFE and FCFS (third subplot of Figure 5). In a single-server queue, it is well known that SJF performs relatively better than FCFS as congestion in the system increases; see Wierman (2007b). However, less is understood about how SJF fares in comparison with FCFS in a multi-server queue with customer abandonment, which is the setting that we consider here. The first subplot in Figure 5 suggests that the same observations as in a single-server setting continue to hold in our more complex setting as well. On the other hand, it is also to be expected that robustness to estimation errors
decreases with the level of congestion. That is, we expect SJFE to perform significantly worse than SJF as congestion in the system increases. This is because the higher the congestion, the longer the queue of waiting customers, and the smaller the probability of selecting the correct customer (with the shortest job), even as the correlation between $S$ and $\hat{S}$ is held fixed. In other words, we observe a tension between the two effects: While SJF should perform better compared to FCFS in a more congested system, SJFE may perform worse because robustness to estimation errors decreases; this corresponds to comparing the first two subplots in Figure 5. Our numerical results (third subplot of Figure 5) illustrate that, in aggregate, the first effect dominates, and SJFE performs relatively better than than FCFS as congestion increases, i.e., we can formulate the following takeaway:

Insight # 1. SJFE leads to smaller queues and shorter overall waiting times, compared to FCFS, as system congestion increases.

\[ \text{Figure 5} \quad \text{Impact of varying the arrival rate } \lambda \text{ on the comparative performance of SJF, SJFE, and FCFS.} \]

\textbf{System Size.} We now turn to studying the effect of increasing system size on the performance of SJFE. We fix the traffic intensity $\rho = 1.2$, and vary both the arrival rate $\lambda$ and the number of servers $N$. In particular, we consider values of the number of servers ranging from 10 to 70, in increments of 10. We deliberately consider small to moderately-sized systems as they are more relevant from a practical perspective. To understand the impact of varying the number of servers, we first study the effect of increasing system size on the performance of SJF compared to FCFS (first subplot of Figure 6). Then, we study the effect of increasing system size on the robustness of SJFE to estimation errors (second subplot of Figure 6), i.e., we compare SJFE and SJF for varying system sizes. Finally, we study the aggregate of these two effects, by comparing SJFE and FCFS, in the third subplot of Figure 6.
We first note that as the size of the system increases, the length of the queue increases as well, even while the traffic intensity is held fixed. Thus, the probability that the first customer in line (picked under FCFS) actually has the shortest service time in queue, decreases. In other words, the expected queue length under FCFS becomes increasingly long relative to SJF. On the other hand, when there are many servers in the system, we expect robustness to estimation errors to increase. Indeed, the effect of a mistake in scheduling in SJF versus SJFE (e.g., due to picking a customer who does not have the shortest service time) should decrease for the same value of correlation, since the effect of that error should be offset by the large number of servers. In other words, the customer who has a long service time and was picked by mistake occupies only one server among many. Thus, there is a tension between those two effects, and it is unclear what the aggregate effect is, i.e., how SJFE fares in comparison to FCFS as the system size increases. Based on our numerical results, illustrated in the third subplot of Figure 6, we can derive the following insight:

Insight # 2. SJFE leads to smaller queues and shorter overall waiting times, compared to FCFS, as the system size increases.

Figure 6  Impact of varying the number of servers $N$ on the comparative performance of SJF, SJFE, and FCFS.

Patience distribution. We now vary the distribution of the time to abandon in the system. In particular, we consider $M/LN/100+GI$ queues with $\lambda = 120$. We consider three abandonment-time distributions: exponential ($M$), Erlang (sum of two exponentials, $E_2$), and hyperexponential with balanced means and squared coefficient of variation equal to 4 ($H_2$). We also consider deterministic times to abandon $D$ in the supplement. Throughout, we let the mean time to abandon, $E[T_{AB}] = 1$.

Our choice of time-to-abandon distributions corresponds to considering alternative shapes for the failure rate of the abandonment distribution. In particular, while the $M$ distribution has a
constant failure rate, the $E_2$ distribution has increasing failure rate (IFR), while the $H_2$ distribution has decreasing failure rate (DFR). This is important to note because it has been observed in the extant literature that the optimal scheduling policy in a multiserver queue with impatient customers depends on the monotonicity of the failure rate of the time to abandon distribution (Bassamboo and Randhawa 2015). We also note that the $H_2$ distribution is more variable than $M$ which, in turn, is more variable than $E_2$, and that there is empirical evidence which substantiates that abandonment times in practice are a good fit for a DFR distribution (Bolandifar et al. 2018).

On one hand, we know that with DFR patience, FCFS is the minimizer of $E[Q]$ among blind policies, i.e., ones that do not assume knowledge of underlying realizations of random processes (Bassamboo and Randhawa 2015). However, with IFR patience, it is LCFS that minimizes $E[Q]$ among blind policies instead. Thus, we expect FCFS to perform better than SJF with abandonment distributions that have DFR, e.g., like $H_2$. On the other hand, with all else constant, DFR patience times lead to less congestion in the system because abandoning customers tend to abandon earlier due to the shape of their abandonment distribution. As we saw earlier, decreased congestion in the system corresponds to more robustness in face of estimation errors, i.e., SJFE should fare better compared to SJF. The aggregate of those two effects, which move in contradicting directions, is unclear. Table 6 illustrates the results of our numerical experiments. In particular:

Insight # 3. SJFE leads to smaller queues and shorter overall waiting times, compared to FCFS, for DFR time-to-abandon distributions.

<table>
<thead>
<tr>
<th>Patience</th>
<th>$E[Q_{FCFS}]$</th>
<th>$E[Q_{SJF}]$</th>
<th>$E[Q_{FCFS}] / E[Q_{SJF}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_2$</td>
<td>6.61</td>
<td>12.8</td>
<td>0.93</td>
</tr>
<tr>
<td>$M$</td>
<td>9.10</td>
<td>20.1</td>
<td>2.20</td>
</tr>
<tr>
<td>$E_2$</td>
<td>11.9</td>
<td>40.1</td>
<td>3.38</td>
</tr>
</tbody>
</table>

Table 6 Impact of varying the time-to-abandon distribution on the comparative performance of SJF, SJFE, and FCFS.
Mean time to abandon. We now turn to studying the impact of varying the mean time to abandon in the system. In particular, we consider the $M/\text{LN}/100 + M$ queueing system with $\lambda = 120$, and consider three values for the mean time to abandon: $E[T_{Ab}] = 0.5, 1, \text{ and } 5$. We report our results in Table 7. First, we note that as the mean time to abandon increases, the congestion in the system increases as well, so that we expect to observe the same effect as when varying the arrival rate in the system. Indeed, on one hand, a more congested system is favorable to SJF over FCFS, and on the other hand, a more congested system is less robust to estimation errors so that SJFE should fare worse than SJF. There is tension between the two effects, and it is unclear what the aggregate effect is. Our numerical experiments, summarized in Table 7, suggest that:

Insight # 4. SJFE leads to smaller queues and shorter overall waiting times, compared to FCFS, for more patient customers.

<table>
<thead>
<tr>
<th>Mean time to abandon</th>
<th>$E[Q_{FCFS}]$</th>
<th>$E[Q_{SJF}]$</th>
<th>$E[Q_{FCFS}]$/$E[Q_{SJF}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>10.2</td>
<td>5.42</td>
<td>1.88</td>
</tr>
<tr>
<td>1</td>
<td>20.3</td>
<td>9.10</td>
<td>2.21</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>29.3</td>
<td>3.41</td>
</tr>
</tbody>
</table>

Table 7 Impact of varying the mean time-to-abandon distribution on the comparative performance of SJF, SJFE, and FCFS.

4.3. Investigating Unfairness: Heuristic Policies

So far, we have focused on comparing the performances of SJFE and FCFS by focusing on the average busyss in the system, as measured by the average queue length (or mean waiting time). In this section, we take a closer look at performance. In particular, we consider three waiting time measures: We let $W$, $[W|S]$ and $[W|A]$ denote the steady-state waiting time until either abandonment or service, the waiting time conditional on being served, and the waiting time conditional on abandoning, respectively. We calculate point estimates of the means and variances of these waiting time measures. We will observe in the figures of this section that, while the SJFE policy is beneficial compared to FCFS in reducing the overall congestion level in the system, i.e., it leads to shorter overall waiting time on average, it also exhibits unfairness towards customers. In particular, when customers are scheduled according to their service-time estimates, it is well known that this may lead
to starving long jobs, i.e., holding them in queue for a long time. Because customers are impatient in our setting, customers who are kept in line for a long time will eventually abandon, however the mean waiting time until abandonment, i.e., $E[W|A]$ increases. On the other hand, the mean waiting until service, $E[W|S]$, decreases, because served customers are those with shorter service requirements, and they are seen sooner than under FCFS scheduling. We consider the $M/LN/100 + M$ queueing system, and vary the correlation between $S$ and $\hat{S}$. We let the traffic intensity $\rho = 1.2$.

That unfairness towards customers with SJF has been extensively studied in the literature (e.g., see Bansal and Harchol-Balter (2001)), albeit strictly in the context of a single-server system with infinitely patient customers. Here, we consider a more complex setting, which is more appropriate to describe real-life service systems. We also propose heuristic policies to limit the effect of unfairness on customers, while not compromising on the superior performance of SJFE relative to FCFS. In particular, we propose the following heuristic. In order to limit the extent to which abandoning customers with longer service times are held in the queue, we propose to keep track of the delay of the longest-waiting-customer in queue (at the head of the line). Once that delay exceeds a certain threshold, e.g., $k \times E[T_{Ab}]$ for some $k \geq 0$, we serve that customer next. Otherwise, we serve customers according to the SJFE policy. We investigate the performance of that heuristic, for various values of $k$, in Figure 7. In particular, we consider $k$ ranging from 0.25 to 2.

In Figure 7, we plot point estimates of $E[W]$, $E[W|S]$ and $E[W|A]$ for various values of $k$. We also include estimates corresponding to both SJFE and FCFS. As anticipated, we observe that both $E[W]$ and $E[W|S]$ are minimized when considering SJFE. We also note that for higher values of $k$, the performance of the heuristic is closer to SJFE because higher values of $k$ correspond to switching to serving the longest waiting customer (i.e., as in FCFS) less often. The third subplot of Figure 7 illustrates the improvement in $E[W|A]$ under the scheduling heuristic, i.e., the improvement in fairness. Ultimately, the choice of $k$ depends on the manager’s preferences in striking a balance between the reduction in the overall congestion of the system (as a result of SJFE) and the improvement of fairness towards customers (by limiting the extent to which customers wait before abandonment).

Finally, we also sought the feedback of call-center managers to gage their tolerance towards the unfairness issues created by the SJF policy. Because call-center performance is typically tracked based on target service levels, which are improved under SJF, we found managers to be enthusiastic about that policy despite the unfairness issue. To mitigate unfairness towards customers, managers were especially open to exploring heuristic approaches which combine the FCFS and SJF policies, as we do in this section. While not conclusive, the initial feedback that we received from managers about the practical implementation of the SJF policy is encouraging.
5. Theoretical Insights in a Simplified Setting

In this section, we investigate the operational performance of the non-preemptive SJF policy with noisy service-time information by using a queueing-theoretic mode of analysis. As such, we deepen our understanding into that performance and formalize some of our earlier numerical observations.

As discussed in §2, the only work that we are aware of which considers a multi-server setting is Grosof et al. (2018). However, Grosof et al. (2018) assumes perfect knowledge of service times, and a preemptive SRPT scheduling policy. Since direct analysis in multi-server queueing models with partial information and customer abandonment is prohibitively difficult and lies beyond the scope of this paper, we focus instead on a simplified single-server setting with infinitely patient customers. Despite the simplicity of the modelling context that we consider, our results are useful to gain insight into performance in more complicated, and more realistic, settings. In this section, we prove that scheduling under noisy service-time information leads to a smaller expected waiting time than any finite-priority class-based scheduling policy, e.g., including FCFS, under a mild condition on the pair of random variables characterising the true and estimated service times. This condition holds for the service-time models of our empirical study.

In §5.1, we describe our model and notation. In §5.2, we derive an expression for the expected waiting time under the SJF policy with noisy service-time information. In §5.3, we prove that, under a mild condition, SJF based on noisy information leads to a shorter expected waiting time than any finite-priority class-based policy. While FCFS is a special case of a finite-priority class-based policy, where we only have one class, we also show that superior performance relative to FCFS can be obtained under an even weaker condition. Finally, we consider the special case of lognormal distributions in §5.4, since that assumption is consistent with the service-time models of this paper.

5.1. Model and Notation

We consider a single-server, $M/G/1$, queueing system. The arrival process is according to a Poisson process with rate $\lambda$, and service times are assumed to be independent and identically distributed
according to a general distribution. To each customer \(c\), we associate a pair of random variables, \((S_c, \hat{S}_c)\), denoting the actual and estimated service times for that customer. We assume that \(S_c\) and \(\hat{S}_c\) have finite second moments. We drop dependence on the customer index when that index does not matter. We let \(f_S\) and \(f_{\hat{S}}\) denote the marginal densities of \(S\) and \(\hat{S}\), respectively. We note that we do not impose a specific functional relationship between \(S\) and \(\hat{S}\).

We define \(E[S] \equiv 1/\mu\), and assume that \(\lambda < \mu\) so that the system is stable, i.e., a proper steady-state distribution exists. For a given scheduling policy \(P\), we let \(W_P(S, \hat{S})\) be a random variable with the steady-state distribution of the waiting time (until service) under scheduling policy \(P\) for actual service time \(S\) and estimated service time \(\hat{S}\). We also let \(W_P(s, \hat{s})\) be a random variable with the steady-state distribution of the waiting time conditional on the event \(S = s\) and \(\hat{S} = \hat{s}\), i.e., it has the same distribution as the random variable \([W_P(s, \hat{s})|S = s, \hat{S} = \hat{s}]\).

### 5.2. SJF with Noisy Service Times

We begin by investigating the performance of the non-preemptive SJF policy with noisy service-time estimates. In other words, we assume that customers are served, in a non-preemptive fashion, in increasing order of their estimated service times, since actual service times are unknown at scheduling epochs. (If the server is idle when a new customer arrives, then that customer is served immediately and remains in service until her service time has elapsed.) To do analysis, we use the “tagged job technique” in Wierman (2007b). That is, we first consider non-preemptive threshold-based policies, and then obtain results for SJF by taking appropriate limits. We assume that there are thresholds \(0 = t_0 < t_1 < ... < t_n = \infty\) such that an arriving customer with estimated service time \(\hat{s}\) is assigned priority \(i\) if, and only if, \(\hat{s} \in [t_{i-1}, t_i)\) for \(0 \leq i \leq n\). We assume that customer classes are served in increasing order, i.e., the lowest customer class has the highest non-preemptive priority. We denote this scheduling policy by \(NP_n\), where \(n\) indexes the number of classes. We assume that a tagged customer is assigned to some class \(k\), where \(1 \leq k \leq n\), i.e., we assume that \(\hat{s} \in [t_{k-1}, t_k)\) for that customer. We can write the expected waiting time for a customer in the \(k\)th class as follows:

\[
E[W_{kNP_n}] = E[W_0] + \sum_{i=1}^{k} E[S1(\hat{S} \in [t_{i-1}, t_i))] \lambda E[W_{iNP_n}] + \sum_{i=1}^{k-1} E[S1(\hat{S} \in [t_{i-1}, t_i))] \lambda E[W_{kNP_n}],
\]

where \(1(A)\) denotes an indicator random variable for a given event \(A\). In (1), \(E[W_0]\) is the remaining work of the customer currently in service, the first sum term denotes the amount of work that the new arriving customer finds in queue upon arrival, and the second sum term denotes the new work,
of higher priority, that arrives during the waiting time of the class-$k$ customer. It is well known that $\mathbb{E}[W_0] \equiv \lambda \mathbb{E}[S^2]/2$. We define the load in customer class $i$ as

$$\rho(t_{i-1}, t_i) \equiv \lambda \mathbb{E}[S 1(\hat{S} \in [t_{i-1}, t_i])].$$

Upon simplification, we obtain from (1) the following system of $n$ equations:

$$\mathbb{E}[W_k^{NP_n}] = \frac{\mathbb{E}[W_0] + \sum_{i=1}^{k-1} \rho(t_{i-1}, t_i) \mathbb{E}[W_i^{NP_n}]}{1 - \sum_{i=1}^{k} \rho(t_{i-1}, t_i)}.$$

That system can be solved recursively to obtain the expected waiting time of the $k^{th}$ class:

$$\mathbb{E}[W_k^{NP_n}] = \frac{\lambda \mathbb{E}[S^2]}{2(1 - \sum_{i=1}^{k-1} \rho(t_{i-1}, t_i))(1 - \sum_{i=1}^{k} \rho(t_{i-1}, t_i))},$$

for $\rho(t_{i-1}, t_i)$ as given in (2). Now, taking the limit as the number of classes $n \to \infty$ and $t_i - t_{i-1} \to 0$:

$$\mathbb{E}[W_{S^{JF}}(s, \hat{s})] = \frac{\lambda \mathbb{E}[S^2]}{2(1 - \rho(\hat{s}))^2},$$

under the SJF policy for a customer with actual and estimated service times $(s, \hat{s})$, where we define $\lim_{t_{i-1}, t_i \to 0} \rho(t_{i-1}, t_i) \equiv \rho(\hat{s}) = \lambda \mathbb{E}[S 1(\hat{S} \leq \hat{s})]$. In what follows, our objective is to compare the operational performance in (3) to other non-preemptive scheduling policies, including FCFS.

5.3. Non-Preemptive Policies with a Finite Number of Priority Classes

To assess the performance of the SJF policy with noisy service-time estimates, we compare it to other class-based policies with a finite number of classes. To be specific, we define an $n-$class priority policy $\pi \in \Pi_n$, where $\Pi_n$ is the set of all such policies, by partitioning $[0, \infty)$ into $n$ disjoint subsets of non-negative reals $I_{j,\pi}$, where $1 \leq j \leq n$. We let customer class $j$ have non-preemptive priority over class $k$ if $j < k$, where a customer $c$ belongs to class $j$ if her estimated service time $\hat{s}_c \in I_{j,\pi}$.

The main result of this section is Theorem 1, where we formulate a sufficient condition on the pair of random variables $(S, \hat{S})$ under which the SJF policy has a superior performance compared to any policy in the set $\Pi_n$. (In this paper, we use “increasing” to mean non-decreasing.) To set the stage for that theorem, we begin with the following definition.

**Definition 1.** Let $X$ and $Y$ denote random variables. We say that $Y$ is stochastically increasing in $X$, written as $SI(Y|X)$, if the following condition holds:

$$\mathbb{P}(Y > y|X = x) \text{ is increasing in } x \text{ for all } y.$$ 

To illustrate, assume that $S = \hat{S} + \epsilon$, where $\hat{S}$ and $\epsilon$ are independent and $\epsilon$ has a positive density function, so that its cumulative distribution function is increasing. Then, it is readily seen that $S$ is stochastically increasing in $\hat{S}$. Indeed, we have that:

$$\mathbb{P}(S > y|\hat{S} = x) = \mathbb{P}(\hat{S} + \epsilon > y|\hat{S} = x) = \mathbb{P}(\epsilon > y - x|\hat{S} = x) = \mathbb{P}(\epsilon > y - x),$$
since \( \hat{S} \) and \( \epsilon \) are independent, which is increasing in \( x \) for fixed \( y \). We note that our service-time models in §3.1 satisfy the relation above, i.e., \( S \) is stochastically increasing in \( \hat{S} \) under those models.

5.3.1. Theorem 1 and Corresponding Lemmas. Theorem 1 states that if \( S \) is stochastically increasing in \( \hat{S} \), as per our service-time models, then the SJF policy outperforms any class-based policy. Our results in this section are in the same spirit as the results of Theorems 1, 2 and 3 in Argon and Ziya (2009), and our proof techniques are similar. However, the setting of that paper is different from ours: In Argon and Ziya (2009), there are two classes of customers, a high-priority class and a low-priority class. Customers send a signal, which is the probability of belonging to the high class. Thus, customer types are uncertain. Argon and Ziya (2009) find that scheduling according to the highest signal first is optimal among the class of finite priority policies.

In our setting, we have a continuum of classes, since scheduling is according to service-time estimates. Each customer has a different actual service time, i.e., can be thought of as a class on her own, and estimated service times can be thought of as signals of the actual service times. However, it is important to emphasize that classifying according to increasing service-time estimates (signals), \( \hat{S} \), may not be optimal in our setting. Indeed, for this optimality result to hold, we need to specify a stochastic order on the pair \( (S, \hat{S}) \); we specify such a stochastic order in Theorem 1.

In order to prove Theorem 1, we need some additional lemmas. We state those lemmas in what follows to highlight the general structure of our proof, but we relegate all technical details to the Appendix B. First, we consider a subset of the class of finite-priority policies, \( \tilde{\Pi}_n \subset \Pi_n \), which is the set of threshold-based policies where \( I_{j,\pi} = [s_{j-1}, s_j) \) for \( 0 \equiv s_0 < s_1 < \cdots < s_{n-1} < s_n \equiv \infty \). In other words, the subset \( \tilde{\Pi}_n \) consists of policies \( \pi \) where the sets \( I_{j,\pi} \), defined earlier, correspond to disjoint subintervals in \([0, \infty)\). We begin by establishing Lemma 1, which allows us to reduce our search of optimal \( n \)-class priority policies to the set of threshold-based policies, \( \tilde{\Pi}_n \). An optimal scheduling policy is one which minimizes the expected waiting time in the system.

**Lemma 1.** We assume that the actual service time, \( S \), is stochastically increasing in the estimated service time, \( \hat{S} \). Then, for a fixed number of classes \( n \geq 2 \), there exists a threshold-based policy \( \pi^* \in \tilde{\Pi}_n \) which yields the smallest expected waiting time among the set of finite \( n \)-class priority policies in \( \Pi_n \).

Based on Lemma 1, our search for an optimal \( n \)-class priority policy reduces to threshold-based priority scheduling policies. Recall that scheduling according to shortest service-time estimates can be thought of as a limiting form of finite threshold-based policies, in particular one where the number of classes increases without bound and the intervals corresponding to the different classes shrink to
zero in size. Thus, in order to establish our main result, we need the following additional lemma
where we prove that a finer partition of threshold-based policies does indeed lead to a superior
performance. Let a policy $\pi \in \tilde{\Pi}_n$. Define a new policy, $\tilde{\pi} \in \tilde{\Pi}_{n+1}$, as follows. For $\tilde{\pi}$, pick any class interval $[l, u)$ for $\pi$, and let $l < s < u$. Define a new partition for $\tilde{\pi}$ such that customers with $\hat{S} \in [l, s)$ have higher non-preemptive priority over customers with $\hat{S} \in [s, u)$, while other classes are kept the same. In other words, policies $\pi$ and $\tilde{\pi}$ coincide, except that one of the classes in $\pi$ is split into two classes under $\tilde{\pi}$, and scheduling under both policies is according to increasing service-time estimates.

In Lemma 2, we prove that the expected waiting time under the new policy $\tilde{\pi}$ is at most as large as the original policy $\pi$. That is, a finer partition of classes leads to superior performance.

**Lemma 2.** Assume that $S$ is stochastically increasing in $\hat{S}$. Then, the expected waiting time under $\tilde{\pi} \in \Pi_{n+1}$, as defined above, is at most as large as under $\pi \in \Pi_n$.

We are now ready to formulate our main result, which shows that scheduling according to noisy service-time estimates can lead to superior performance compared to any finite class priority policy.

**Theorem 1.** If $S$ is stochastically increasing in $\hat{S}$, then the expected waiting time under the non-preemptive SJF policy is at most as large as any finite-class non-preemptive priority policy.

It is important to note that the FCFS policy is a special case of a class-based policy (where there is only one customer class). Thus, under the condition in Theorem 1, SJF yields smaller expected waiting time than FCFS as well. However, in this particular case, we can formulate a weaker condition under which SJF is superior to FCFS, as we show next. We let $\mathcal{F}(f_S, f_{\hat{S}})$ denote the set of all bivariate distributions with the same marginal distributions $f_S$ and $f_{\hat{S}}$. Recall that Positive Quadrant Dependence (PQD) is defined as follows (Shaked and Shanthikumar 2007):

**Definition 2.** A random vector $(X_1, X_2)$ with joint distribution $F$ and marginal cumulative distributions $F_1$ and $F_2$ is said to be positive quadrant dependent (PQD) if

$$F(x_1, x_2) \geq F_1(x_1)F_2(x_2) \quad \text{for all } x_1 \text{ and } x_2.$$  

Let $(X_1, X_2)$ and $(Y_1, Y_2)$ with joint cdf’s $G$ and $H$, respectively, have the same marginal densities i.e., $(X_1, X_2)$ and $(Y_1, Y_2) \in \mathcal{F}(f_1, f_2)$. Then,

$$(X_1, X_2) \leq_{PQD} (Y_1, Y_2) \quad \text{if, and only if,} \quad G(x_1, x_2) \leq H(x_1, x_2) \quad \text{for all } (x_1, x_2).$$  

Similarly, we say that $(X_1, X_2)$ is negative quadrant dependent (NQD) if

$$F(x_1, x_2) \leq F_1(x_1)F_2(x_2) \quad \text{for all } x_1 \text{ and } x_2.$$  

It is well known that if $S$ is stochastically increasing in $\hat{S}$, then $(S, \hat{S})$ is PQD; e.g., see p. 267 of Lai and Xie (2006). In other words, the sufficient condition in Theorem 1 is stronger than PQD. The following lemma, which compares the performances of SJF and FCFS, holds.

**Lemma 3.** Let $(S, \hat{S}) \in \mathcal{F}(f_S, f_{\hat{S}})$. The following hold:

- If $S$ and $\hat{S}$ are independent, then $\mathbb{E}[W^{SJF}(S, \hat{S})] = \mathbb{E}[W^{FCFS}(S, \hat{S})]$.
- If $(S, \hat{S})$ is PQD, then $\mathbb{E}[W^{SJF}(S, \hat{S})] \leq \mathbb{E}[W^{FCFS}(S, \hat{S})]$.
- If $(S, \hat{S})$ is NQD, then $\mathbb{E}[W^{SJF}(S, \hat{S})] \geq \mathbb{E}[W^{FCFS}(S, \hat{S})]$.

The analytical results that we have described so far allow the comparison of SJF to other scheduling policies. It is also of interest to compare the operational performance of SJF policies based on various service-time models, which yield different service-time estimates. We do so in what follows, by considering the special case of lognormally-distributed service times. We note that with lognormally-distributed service times, the PQD condition in Lemma 1 reduces to the familiar positive correlation assumption between the pair of actual and estimated service times. Thus, our numerical results in §4 are useful in quantifying the extent of improvement in operational performance, of SJF compared to FCFS, as a function of that correlation. We also note that the lognormal distribution is a remarkably good fit to the empirical distribution of service times in our data set.

### 5.4. Lognormal Distributions

In this section, we consider a bivariate lognormal model. In particular, we define $Z \equiv \ln(S)$ and $\hat{Z} \equiv \ln(\hat{S})$. We now consider two alternative service-time models, for the log-transformed service times, yielding normally-distributed service-time estimates with different distributions. The objective of this section is to compare operational performances obtained when scheduling according to SJF and estimates from each service-time model. We assume that $\mathbb{E}[Z] = \theta, \text{Var}[Z] = \sigma_Z^2, \mathbb{E}[\hat{Z}_i] = \hat{\theta}_i, \text{Var}[\hat{Z}_i] = \sigma_{\hat{Z}_i}^2$, where $i = 1, 2$ indexes the corresponding service-time model. We also assume that $(Z, \hat{Z}_1)$ and $(Z, \hat{Z}_2)$ each follow a bivariate normal distribution. We can then prove the following lemma, where we define $r[X, Y]$ as the correlation between $X$ and $Y$.

**Lemma 4.** If $r[Z, \hat{Z}_1] \geq r[Z, \hat{Z}_2]$, then $\mathbb{E}[W^{SJF}(S, \hat{S}_1)] \leq \mathbb{E}[W^{SJF}(S, \hat{S}_2)]$.

Lemma 4 relates operational performance when scheduling according to estimates from $\hat{S}_1$ and $\hat{S}_2$ or, equivalently, $\hat{Z}_1$ and $\hat{Z}_2$. It states a simple condition on the estimates under which SJF scheduling will yield superior performance. In particular, if the correlation between the service-time estimates under model 1 and actual service times is larger than under model 2, then SJF scheduling using estimates from model 1 will yield smaller expected waiting time. Lemma 4 provides theoretical justification to our numerical observations in the previous section, where we saw that higher correlations correspond to a better performance of the SJF policy.
6. Conclusions

In this work, we study the operational performance of personalized scheduling policies with noisy information, using a comprehensive approach combining empirical analysis, simulation modelling, and queueing theory. In particular, we consider the shortest-job-first policy with noisy service-time estimates, SJFE. In our empirical analyses, we show that customers’ contact history information can improve the performance of service-time prediction models significantly. However, even after utilizing all information in the data including time fixed effects, agent fixed effects, and customer contact information, the $R^2$ of the service time prediction models remains quite low. In our simulation studies, we show that if we implement the shortest job first policy using this extremely noisy information, then the impact on the call center’s performance measures may still be considerable. We also perform a comprehensive series of simulation studies to investigate the impact of SJFE using different arrival rates, system sizes, and customer patience levels. Our simulation studies show that scheduling with noisy estimates leads e.g., to shorter waiting times for higher congestion levels, higher customer patience levels, and larger systems. Finally, we consider a stylized single-server queueing model with non-preemptive SJF, and formulate sufficient conditions under which that policy performs better than any finite-class based policy, e.g. including FCFS.

Even though our theoretical results are not limited to the call center setting, our empirical analysis is based on a call center data set. It would be interesting to study the accuracy of service-time predictions and the operational performance of the SJFE policy for other settings. Our theoretical results are provided for a single-server setting without customer abandonment. Future research may also examine the SJFE policy for multiserver settings with customer abandonment. While direct analysis may be prohibitively difficult, relying on approximations, e.g., fluid approximations as in Mandelbaum and Momčilović (2017) seems promising. Finally, we have focused in this paper on personalized scheduling based solely on service-time information. It would be interesting to explore scheduling policies that exploit both service and patience-time information for customers. In particular, one can think of hybrid policies that combine both types of information in an effective manner. Studying the impact of noisy information in such settings remains a topic for future research.

Acknowledgments

We thank the SEELAB at the Technion, especially Professor Avishai Mandelbaum, for generously providing our data set.

References


Appendix A: Comparing the Predictive Power of Regression Models with Machine Learning Models

In this section, we explore the performance of the decision tree and the neural network models in service-time prediction, and show that even though these more complicated models provide slightly better predictive performance, their performance is not significantly better than the regression models introduced in §3.1.1.

We choose the decision tree and the neural network models because these models are on the two extremes of machine-learning prediction models. To be more specific, the decision tree model is a non-pragmatic model that is based on separating observations into subgroups by creating splits on the set of covariates (?). While the neural network is a flexible parametric model that creates new covariates by combining the observables in the data and then models the target variable as a nonlinear function of these covariates (?).

Consistent with the performance evaluation procedure for the regression models explained in Section 3.1.2, we run six out-of-sample tests using the regression tree and the neural network models. In all of these tests, we choose the logarithm of the service time as the target variable and the set of TF+AF+RT+DH+AH+CF variables as the covariates. Given that both the regression tree and the neural network models automatically select the features to maximize the prediction performance in the out-of-sample set, there is no need to exclude any of the covariates when running the tests. For the neural network method, we choose a model
with one layer and six nodes (3 linear and 3 tanH).

Table 8 and 9 show the comparison between the $R^2$s and the RMSEs of the regression model 3', which was chosen as the best model in Section 3.1.3, and the machine learning models, respectively. As can be seen in these tables, both the regression tree and the neural network models perform slightly better than the regression model 3’ with the regression tree performing better than the neural network model. However, the improvement in $R^2$ is less than 1% and the improvement in RMSE is less than 0.1 while the regression model, Model 3’, is much simpler and parsimonious than both machine learning models.

<table>
<thead>
<tr>
<th>$R^2$</th>
<th>Model 3' Regression Tree</th>
<th>Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>8.32%</td>
<td>8.79%</td>
</tr>
<tr>
<td>Test 2</td>
<td>10.19%</td>
<td>10.12%</td>
</tr>
<tr>
<td>Test 3</td>
<td>9.05%</td>
<td>8.83%</td>
</tr>
<tr>
<td>Test 4</td>
<td>5.26%</td>
<td>7.76%</td>
</tr>
<tr>
<td>Test 5</td>
<td>9.28%</td>
<td>9.84%</td>
</tr>
<tr>
<td>Test 6</td>
<td>8.67%</td>
<td>9.01%</td>
</tr>
<tr>
<td>Average</td>
<td>8.46%</td>
<td>9.06%</td>
</tr>
</tbody>
</table>

Table 8 $R^2$s of Model 3' (regression), and the regression tree and neural network models (machine learning).

<table>
<thead>
<tr>
<th>RMSE</th>
<th>Model 3' Regression Tree</th>
<th>Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>0.8208</td>
<td>0.8186</td>
</tr>
<tr>
<td>Test 2</td>
<td>0.8248</td>
<td>0.8251</td>
</tr>
<tr>
<td>Test 3</td>
<td>0.8227</td>
<td>0.8230</td>
</tr>
<tr>
<td>Test 4</td>
<td>0.8448</td>
<td>0.8331</td>
</tr>
<tr>
<td>Test 5</td>
<td>0.8130</td>
<td>0.8099</td>
</tr>
<tr>
<td>Test 6</td>
<td>0.8142</td>
<td>0.8126</td>
</tr>
<tr>
<td>Average</td>
<td>0.8234</td>
<td>0.8204</td>
</tr>
</tbody>
</table>

Table 9 RMSEs of Model 3' (regression), and the regression tree and neural network models (machine learning).

Appendix B: Proofs of Results in §5

B.1. A Useful Technical Lemma

We begin by proving the following result which will be useful for subsequent lemmas.

**Lemma 5.** If $Y$ is stochastically increasing in $X$, then for all $a \leq b \leq c \leq d$,

$$P(X \geq x | a \leq Y \leq b) \leq P(X \geq x | c \leq Y \leq d).$$

Thus, $E[X|a \leq Y \leq b] \leq E[X|c \leq Y \leq d]$.

**Proof.**

$$P(X \geq x | a \leq Y \leq b) = \int_{a}^{b} P(X \geq x | a \leq Y \leq b, Y = y) f_Y(y | a \leq Y \leq b) dy = \int_{a}^{b} P(X \geq x | Y = y) f_Y(y | a \leq Y \leq b) dy \leq \int_{a}^{b} P(X \geq x | Y = b) f_Y(y | a \leq Y \leq b) dy = P(X \geq x | Y = b) \leq P(X \geq x | Y = c)$$
and that we can write the expected waiting time for class policies in threshold-based policy the specified necessary condition. Thus, Theorem 1 of the paper is proved.

Theorem 1. Consider a policy \( \pi \in \tilde{\Pi} \). Then, for fixed \( n \geq 2 \), there exists a threshold-based policy \( \pi^* \in \tilde{\Pi}_n \) which yields the smallest expected waiting time among the set of finite-priority policies in \( \Pi_n \).

Proof. We recall from (2) that for thresholds \( t_0 = 0 \leq t_1 \leq t_2 \leq \ldots \leq t_{n-1} \leq t_n = \infty \),

\[
\rho_{j-1,t_j} = \mathbb{E}[S1(\hat{S} \in [t_{j-1}, t_j))],
\]

and that we can write the expected waiting time for class \( j \) as:

\[
\mathbb{E}[W_{j,\pi}] = \frac{\lambda \mathbb{E}[S^2]}{2(1 - \sum_{i=1}^{l-1} \rho(t_{i-1}, t_i))(1 - \sum_{i=1}^{l} \rho(t_{i-1}, t_i))}.
\]

Consider a policy \( \pi \in \Pi^n \) that is not a threshold-based policy. We will prove that there exists \( \pi^* \in \Pi^n \) which is threshold-based and yields a expected waiting time that is at most as large as \( \pi \).

For this, consider \( \pi \). Then, there must exist an interval \( (l, u) \) and a point \( s \in (l, u) \) such that we define class \( m \) for \( \hat{S} \in [l, s) \) and class \( n \) for \( \hat{S} \in [s, u) \) and \( n \) has non-preemptive priority over \( m \). (We can repeat the same argument for any such interval \( [l, u] \).) Note that for class \( m \), \( \rho(l, s) = \mathbb{E}[S1(\hat{S} \in [l, s])] \) is increasing in \( s \) and has values continuously in \( [0, \mathbb{E}[S1(\hat{S} \in [l, u])] \). Thus, there exists \( \hat{s} \in [l, u] \) such that \( \rho(l, \hat{s}) = \rho(s, u) \).

Let us define a new policy \( \pi' \) such that: If \( \hat{S} \in (l, \hat{s}) \), then customer belongs to class \( n \), and if \( \hat{S} \in [\hat{s}, u) \), then customer belongs to class \( m \). Moreover, class \( n \) has priority over class \( m \) under policy \( \pi' \). Since \( \rho(\hat{s}, u) = \rho(l, s) \)
and $\rho(l, \tilde{s}) = \rho(s, u)$, it must be, using (4), that $\mathbb{E}[W_{m, n}] = \mathbb{E}[W_{m, \tilde{n}}] = \mathbb{E}[W_{n}]$ and $\mathbb{E}[W_{n}] = \mathbb{E}[W_{n, \tilde{n}}]$ (since all other priority classes are unchanged).

To show that $\pi'$ yields an expected waiting time that is at most as large as under $\pi$, we need to show that:

$$\lambda \mathbb{P}(\hat{S} \in (l, \tilde{s}))\mathbb{E}[W_{n, \tilde{n}}] + \lambda \mathbb{P}(\hat{S} \in (s, u))\mathbb{E}[W_{n, \tilde{n}}] \leq \lambda \mathbb{P}(\hat{S} \in (l, s))\mathbb{E}[W_{m, n}] + \lambda \mathbb{P}(\hat{S} \in (s, u))\mathbb{E}[W_{m, n}]. \tag{5}$$

To obtain this, we consider two cases: (i) $s < \tilde{s}$ and (ii) $s \geq \tilde{s}$.

**Case (i).** $s < \tilde{s}$. We write the difference between the RHS and the LHS of (5) as:

$$\mathbb{P}(\hat{S} \in (l, s))\mathbb{E}[W_{m, n}] + \mathbb{P}(\hat{S} \in (s, \tilde{s}))\mathbb{E}[W_{n, \tilde{n}}] + \mathbb{P}(\hat{S} \in (\tilde{s}, u))\mathbb{E}[W_{n, \tilde{n}}] - \mathbb{P}(\hat{S} \in (l, s))\mathbb{E}[W_{m, n}] - \mathbb{P}(\hat{S} \in (s, \tilde{s}))\mathbb{E}[W_{n, \tilde{n}}] - \mathbb{P}(\hat{S} \in (\tilde{s}, u))\mathbb{E}[W_{n, \tilde{n}}]$$

$$= (\mathbb{E}[W_{m}] - \mathbb{E}[W_{n}])\mathbb{P}(\hat{S} \in (l, s)) - \mathbb{P}(\hat{S} \in (\tilde{s}, u))).$$

Recall that $\mathbb{E}[W_{m}] \geq \mathbb{E}[W_{n}]$ since class $m$ has lower priority than class $n$. Also recall that by construction:

$$\rho(l, s) = \mathbb{E}[S1(\hat{S} \in (l, s))] = \mathbb{E}[S1(\hat{S} \in (\tilde{s}, u))] = \rho(s, u).$$

But, by assumption, $S$ is stochastically increasing in $\hat{S}$, so that: $\mathbb{E}[S|\hat{S} \in (l, s)] \leq \mathbb{E}[S|\hat{S} \in (\tilde{s}, u)]$ by Lemma 5. Since

$$\mathbb{E}[S1(\hat{S} \in (l, s))] = \mathbb{E}[S|\hat{S} \in (l, s)]\mathbb{P}(\hat{S} \in (l, s)) = \mathbb{E}[S|\hat{S} \in (\tilde{s}, u)]\mathbb{P}(\hat{S} \in (\tilde{s}, u))$$

we have that $\mathbb{P}(\hat{S} \in (l, s)) \geq \mathbb{P}(\hat{S} \in (\tilde{s}, u))$, so that

$$(\mathbb{E}[W_{m}] - \mathbb{E}[W_{n}])\mathbb{P}(\hat{S} \in (l, s)) - \mathbb{P}(\hat{S} \in (\tilde{s}, u))) \geq 0,$$

as desired.

**Case (ii).** $s \geq \tilde{s}$. We proceed similarly and write the difference between the LHS and the RHS of (5) as:

$$\mathbb{P}(\hat{S} \in (l, s))\mathbb{E}[W_{m, n}] + \mathbb{P}(\hat{S} \in (s, \tilde{s}))\mathbb{E}[W_{n, \tilde{n}}] + \mathbb{P}(\hat{S} \in (\tilde{s}, u))\mathbb{E}[W_{n, \tilde{n}}] - \mathbb{P}(\hat{S} \in (l, s))\mathbb{E}[W_{m, n}] - \mathbb{P}(\hat{S} \in (s, \tilde{s}))\mathbb{E}[W_{n, \tilde{n}}] - \mathbb{P}(\hat{S} \in (\tilde{s}, u))\mathbb{E}[W_{n, \tilde{n}}]$$

$$= (\mathbb{E}[W_{m}] - \mathbb{E}[W_{n}])\mathbb{P}(\hat{S} \in (l, s)) - \mathbb{P}(\hat{S} \in (s, u))).$$

Now, noting that $\rho(l, \tilde{s}) = \mathbb{E}[S1(\hat{S} \in (l, s))] = \mathbb{E}[S1(\hat{S} \in (s, u))] = \rho(s, u)$ and that $S$ is stochastically increasing in $\hat{S}$ so that $\mathbb{E}[S|\hat{S} \in (l, s)] < \mathbb{E}[S|\hat{S} \in (s, u)]$. Thus,

$$\mathbb{P}(\hat{S} \in (l, s)) \geq \mathbb{P}(\hat{S} \in (s, u)),$$

and the desired holds.

Next, we investigate the effect of increasing the number of classes.

**Lemma 7.** Assume that $S$ is stochastically increasing in $\hat{S}$. Let $\pi \in \Pi_n$. Pick any class interval $[l, u)$ and partition it such that a customer with $\tilde{S} \in [l, s)$ has higher non-preemptive priority over a customer for whom $\hat{S} \in [s, u)$, while other classes remain the same. Then, the expected waiting time under the new policy is at most as large as the original policy.
Proof. Consider any interval \((l, u)\) and partition as in the lemma. Let \(\pi\) denote the old policy and \(\pi'\) denote the updated policy. Assume that if \(\hat{S} \in (l, u)\) then this is class \(m\) under \(\pi\), split into classes \(m\) \((\hat{S} \in (l, s))\) and \(m+1\) \((\hat{S} \in (s, u))\) under policy \(\pi'\). We note that the only difference between the two policies is the expected waiting time in the interval \((l, u)\). Let \(0 = s_0 < s_1 < \cdots < s_{m-1} \equiv l < s_m \equiv s < u \equiv s_{m+1}\) denote the thresholds of classes. Recall that, under \(\pi'\): \((l, s_m)\) corresponds to class \(m\) and \((s_m, s_{m+1})\) corresponds to class \(m+1\). On the other hand, under \(\pi\): \((l, s_{m+1})\) corresponds to class \(m\). Then, we need to show that:

\[
\lambda \mathbb{P}(\hat{S} \in (s, u)) \cdot (\mathbb{E}[W_{m+1, u'}] - \mathbb{E}[W_{m, u}]) + \mathbb{P}(\hat{S} \in (l, s)) \cdot (\mathbb{E}[W_{m, u'}] - \mathbb{E}[W_{m, u}]) \leq 0. \tag{6}
\]

Note that:

\[
\mathbb{E}[W_{m, u}] - \mathbb{E}[W_{m, u'}] = \frac{\lambda \rho(s, u)}{1 - \lambda \sum_{k=1}^{m+1} \rho(s_{k-1}, s_k)} \mathbb{E}[W_{m, u'}],
\]

\[
\mathbb{E}[W_{m+1, u'}] - \mathbb{E}[W_{m, u}] = \frac{\lambda \rho(l, s)}{1 - \lambda \sum_{k=1}^{m+1} \rho(s_{k-1}, s_k)} \mathbb{E}[W_{m, u'}].
\]

Thus, the sign of (6) is the same as the sign of:

\[
\mathbb{P}(\hat{S} \in (s, u)) \mathbb{P}(\hat{S} \in (l, s)) [\mathbb{E}[S|\hat{S} \in (l, s)] - \mathbb{E}[S|\hat{S} \in (s, u)]].
\]

Since \(\mathbb{E}[S|\hat{S} \in (l, s)] \leq \mathbb{E}[S|\hat{S} \in (s, u)]\) by the stochastically increasing property of \(S\) in \(\hat{S}\), the desired follows.

Finally, we can show that classifying customers in increasing order of their predicted service times is optimal among the set of all finite-priority policies.

**Lemma 8.** The expected waiting time under the non-preemptive shortest predicted job first policy is at most as large as any finite-class-priority policy.

**Proof.** The proof follows along the lines of the argument in Theorem 3 in Argon and Ziya (2009), so we omit the corresponding details.

Combining the results above establishes Theorem 1 of the paper.

### B.3. Proof of Lemma 3

**Proof.**

• To prove the first part, we assume that \(S\) and \(\hat{S}\) are independent. We then use a coupling argument.

Consider two queueing systems where we couple all arrivals and service times. In system I, serve customers according to a first-come-first-served discipline. In system II, serve customers according to SJF as follows: whenever customer \(C\) is served in system I, i.e., is head-of-line, with \(n\) customers in queue, generate random variates \(\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n\) and assign \(\hat{s}_n = \min\{\hat{s}_1, \ldots, \hat{s}_n\}\) to customer \(C\). This can be done without loss of generality since service times and their estimates are independent. Then, customer \(C\) is picked to be served in system II as well. We proceed similarly for all customers, so that the two systems have sample paths which are perfectly coupled. Thus, \(\mathbb{E}[W_{S\text{JF}}(S, \hat{S})] = \mathbb{E}[W_{\text{FCFS}}(S, \hat{S})].\)
To prove the second part, let \((S, \tilde{S})\) denote a random vector with the same marginal distributions as \((S, \hat{S})\) but where \(S\) and \(\tilde{S}\) are independent. We begin by showing that if \((S_1, \hat{S}_1) \leq_{PQD} (S_2, \hat{S}_2)\) then

\[
E[W_{SJF}^{S}(S_1, \hat{S}_1)] \geq E[W_{SJF}^{S}(S_2, \hat{S}_2)].
\]

To do so, we proceed as follows. Let \(S_1, S_2 \sim S\) and \(\hat{S}_1, \hat{S}_2 \sim \hat{S}\). We can write:

\[
E[W_{SJF}^{S}(S_1, \hat{S}_1)] = \int_{0}^{\infty} \frac{\lambda E[S^2]}{2(1 - \rho(x))^2} f_{\hat{S}_1}(x) dx
\]

\[
= \frac{2}{\lambda E[S^2]} \int_{0}^{\infty} \frac{1}{1 - \lambda E[S_1(\hat{S}_1 \leq x)]} f_{\hat{S}_1}(x) dx
\]

\[
= \frac{2}{\lambda E[S^2]} \int_{0}^{\infty} \frac{1}{1 - \lambda \int_{0}^{\infty} P(S_1(\hat{S}_1 \leq x)) dy} f_{\hat{S}_1}(x) dx
\]

\[
= \frac{2}{\lambda E[S^2]} \int_{0}^{\infty} \frac{1}{1 - \lambda \int_{0}^{\infty} P(S_1 \leq y, \hat{S}_1 \leq x) dy} f_{\hat{S}_1}(x) dx
\]

\[
= \frac{2}{\lambda E[S^2]} \int_{0}^{\infty} \frac{1}{1 - \lambda E[S] + \lambda \int_{0}^{\infty} P(S_1 \leq y, \hat{S}_1 \leq x) dy} f_{\hat{S}_1}(x) dx
\]

\[
\leq \frac{2}{\lambda E[S^2]} \int_{0}^{\infty} \frac{1}{1 - \lambda E[S] + \lambda \int_{0}^{\infty} P(S_0 \leq y, \hat{S}_0 \leq x) dy} f_{\hat{S}_1}(x) dx
\]

\[
= E[W_{SJF}^{S}(S_2, \hat{S}_2)].
\]

Then, by independence: \(E[W_{FCFS}^{S}(S, \hat{S})] = E[W_{FCFS}^{S}(S, \hat{S})]\). But, we also have that \((S, \hat{S}) \leq_{PQD} (S, \hat{S})\).

Thus, by the above, it must hold that:

\[
E[W_{SJF}^{S}(S, \hat{S})] \leq E[W_{SJF}^{S}(S, \hat{S})] = E[W_{FCFS}^{S}(S, \hat{S})] = E[W_{FCFS}^{S}(S, \hat{S})].
\]

The proof proceeds as above, by reversing the corresponding inequalities.

\[\blacksquare\]

### B.4. Lemma 4: Additional Results and Proof

#### B.4.1. Additional results

We begin by establishing the following results. Let \(r[X, Y]\) denote the correlation between random variables \(X\) and \(Y\). We assume that \((S, \hat{S}) \sim (e^Z, e^Z)\) where \((Z, \hat{Z})\) is a bivariate normal pair with \(E[Z] = \theta_1\) and \(E[\hat{Z}] = \theta_2\) and \(Var[Z] = \sigma_1^2\) and \(Var[\hat{Z}] = \sigma_2^2\). We let

\[
\text{Cov}[Z, \hat{Z}] = \alpha \sigma_1 \sigma_2,
\]

so that \(\alpha\) measures the level of information contained in the estimated service times. That is, the correlation is given by \(r[Z, \hat{Z}] = \alpha\). The following lemma holds.

**Lemma 9.** Consider two pairs \((Z_1, \hat{Z}_1)\) and \((Z_2, \hat{Z}_2)\), where \(Z_1\) and \(Z_2\) have the same marginal distribution, and \(\hat{Z}_1\), \(\hat{Z}_2\) have the same marginal distribution.

\[\text{If } r[Z_1, \hat{Z}_1] \leq r[Z_2, \hat{Z}_2] \text{ then } (S_1, \hat{S}_1) \leq_{PQD} (S_2, \hat{S}_2).\]
Proof. By Lemma 3 of Wu et al. (2018), it holds that \((Z_1, \hat{Z}_1) \leq_{PQD} (Z_2, \hat{Z}_2)\). Since the PQD order is preserved under monotonically increasing transformations, we must also have that: \((S_1, \hat{S}_1) \leq_{PQD} (S_2, \hat{S}_2)\). ■

The following corollary is immediate by combining Lemma 9 and Lemma 3.

**Corollary 1.** For \(\alpha \in [-1, 1]\) in (7), let \(W^{SJF}(\alpha)\) denote the steady-state waiting time under the SJF discipline with estimated service times \(\hat{S}\). Then, \(\mathbb{E}[W^{SJF}(\alpha)]\) is monotonically decreasing in \(\alpha\), and we have that:

\[
\begin{align*}
\mathbb{E}[W^{SJF}(-1)] & \geq \mathbb{E}[W^{FCFS}], \\
\mathbb{E}[W^{SJF}(0)] & = \mathbb{E}[W^{FCFS}], \\
\mathbb{E}[W^{SJF}(1)] & \leq \mathbb{E}[W^{FCFS}].
\end{align*}
\]

(B.4.2. Proof of Lemma 4. Note that linear transformations preserve bivariate normality so

\[
\left( Z, \frac{\hat{Z}_i - \mathbb{E}[\hat{Z}_i]}{\sigma_{\hat{Z}_i}} \right)
\]

has also bivariate normal distribution for \(i = 1, 2\). We let \(\hat{X}_i \equiv \frac{\hat{Z}_i - \mathbb{E}[\hat{Z}_i]}{\sigma_{\hat{Z}_i}}\). For these transformed pairs, we have the same marginal distributions for \(i = 1, 2\). Indeed, \(\hat{X}_i\) has a standard normal distribution for \(i = 1, 2\). Thus, we can apply Lemma 9 above. In particular,

If \(r[Z_1, \hat{X}_1] \leq r[Z_2, \hat{X}_2]\), then \((Z_1, \hat{X}_1) \leq_{PQD} (Z_2, \hat{X}_2)\),

where \(Z_1\) and \(Z_2\) both have the same distribution as \(Z\). This also implies that \((e^{Z_1}, e^{\hat{X}_1}) \leq_{PQD} (e^{Z_2}, e^{\hat{X}_2})\). By Lemma 3, this implies that \(\mathbb{E}[W^{SJF}(e^{Z_1}, e^{\hat{X}_1})] \geq \mathbb{E}[W^{SJF}(e^{Z_2}, e^{\hat{X}_2})]\).

However, note that:

\[ r[Z, \hat{X}_i] = \frac{\text{Cov}[Z, \hat{X}_i]}{\sigma_Z \sigma_{\hat{X}_i}} = \frac{\text{Cov}[Z, \hat{Z}_i]}{\sigma_Z \sigma_{\hat{Z}_i}} = r[Z, \hat{Z}_i], \]

and, importantly, we also have that

\[ \mathbb{E}[W^{SJF}(e^{Z_i}, e^{\hat{X}_i})] = \mathbb{E}[W^{SJF}(e^{Z_i}, e^{\hat{Z}_i})], \]

since imposing a deterministic and increasing transformation on the predictions \(\hat{Z}_i\) (to obtain \(\hat{X}_i\)) does not affect the ranking of the customers. Thus, we obtain that:

If \(r[Z, \hat{Z}_1] \leq (\geq) r[Z, \hat{Z}_2]\), then \(\mathbb{E}[W^{SJF}(e^{Z}, e^{\hat{X}_1})] \geq (\leq) \mathbb{E}[W^{SJF}(e^{Z}, e^{\hat{X}_2})]\),

as desired.