Impact of Queue Configuration on Service Time: Evidence from a Supermarket

Jingqi Wang
Faculty of Business and Economics, the University of Hong Kong, Hong Kong, jingqi@hku.hk
Yong-Pin Zhou
Michael G. Foster School of Business, University of Washington, Seattle, WA 98195, yongpin@uw.edu

We study how queue configuration affects human servers' service time by comparing dedicated with shared queues using field data from a natural experiment in a supermarket. We hypothesize that queue configuration may affect servers' service rate through several mechanisms: pooling may affects service rate directly due to social loafing effect and competition effect, and indirectly via its impact on queue length. To investigate these impacts, we take advantage of the supermarket's checkout layout, and use a data set containing both checkout transaction details and queue information collected from video recordings in the supermarket. After we control for the queue length, we find that servers in dedicated queues are about 10.7% faster than those in shared queues, mainly due to the social loafing effect. We also demonstrate that pooling has an indirect negative effect on service time through its impact on queue length. In addition, the queue configuration's direct effect and its indirect queue length effect function independently to each other. In aggregation, the social loafing effect dominates, and servers slow down (a 6.86% increase in service time) in shared queues.

Key words: server behavior, empirical operations management, social loafing in queues, queueing, pooling

1. Introduction

In most developed countries, the service sector accounts for more than 70% of the GDP (The World Bank 2016). As direct labor costs can reach 60% to 70% of a service firm's total operating costs (Tan and Netessine 2014), managing an appropriate level of staffing

becomes a paramount issue. High staffing levels are costly, but low staffing levels can lead to long waiting times, low customer satisfaction, and even lost sales. Having servers use a shared queue to serve all the customers has been considered as one way to reduce the average waiting time without increasing staffing levels, due to its pooling benefit. However, as we shall show, this result critically depends on how queue configuration affects servers' service rate. If servers work much more slowly in shared queues than in dedicated queues, then shared queues could lead to longer average waiting time with the same staffing level. As a result, it is important to understand servers' behavior when deciding the queue configuration.

In addition, most widely used queueing models in both academic literature and practice build upon the assumption of exogenous service rate (Brockmeyer et al. 1948) which, while plausible for non-human servers, is problematic for human servers. According to recent research, human service rate can be affected by operational environment factors, such as queue configuration (e.g., Song et al. 2015, Shunko et al. 2014), workload (e.g., Kc and Terwiesch 2009, Jaeker et al. 2012), and deadline (e.g., Deo et al. 2014). For a broader perspective on behavioral issues in operations management, please see Bendoly et al. (2010).

In this paper, we examine the effect of queue configuration on human servers' service time. Two common queue configurations exist: in the first, one dedicated queue leads to each server; in the second, queues are pooled, and each shared queue leads to multiple servers. *All else being equal*, the pooling of queues is long considered a sure way to reduce labor costs without sacrificing waiting time (Kleinrock 1976), but Song et al. (2015) and Shunko et al. (2014) suggest that all else may not be equal. Rothkopf and Rech (1987) conjecture that pooling may lead to longer service times. In this paper, we investigate this conjecture by testing effects that are related to queue configuration, as summarized in Figure 1.

Queue configuration could impact a human server's service time in two ways. First, the impact can be direct: on the one hand, the server may slow down when serving a shared queue to reduce her share of the work and the effort required to perform this work. This is known as the the social loafing effect. On the other hand, when the server receives

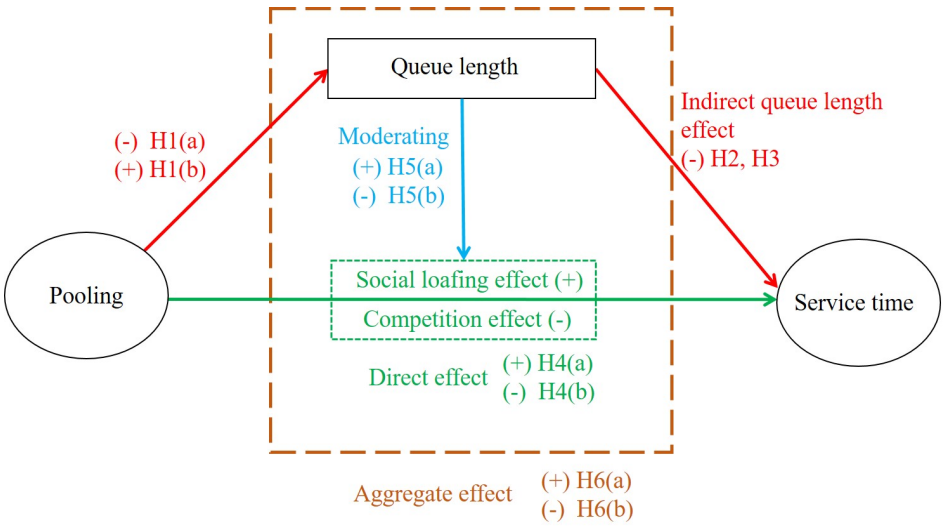

Figure 1 Summary of Hypotheses

a throughput-based bonus, she may speed up in a shared queue in order to get more work and, consequently, more pay. This is known as the competition effect. We call the sum of these two effects the *direct effect* of queue configuration on service time. Because social loafing and competition affect service time in opposite directions, the direction of their aggregate impact on service time (i.e., the direct effect) is unclear. Second, queue configuration can also indirectly impact a human server's service time by affecting queue length that the server faces. Pooling may affect the queue length, and human servers' response to increased workload could vary (Delasay et al. 2016), depending on the nature of the service and the particular industry.

In our paper, we examine both direct and indirect effects. We empirically test different mechanisms through which queue configuration can affect servers' service time, and study the resulting implications on the design of queueing systems.

In the empirical analysis, we use a data set from a supermarket in Shanghai, China. We collect detailed transaction records from the supermarket's IT system, and extract timing information from video recordings that cover the checkout area. In addition, the supermarket places both dedicated queues and shared queues next to each other, in an alternating pattern, which allows us to identify the impact of queue configuration.

Using this data set, we find that the pooling of queues has significant direct and indirect effects on servers' service time. Indirectly, shared queues are longer, which causes servers to speed up. Directly, the social loafing effect is stronger than the competition effect when we control for the queue length, which leads to a total direct effect of a 10.7% increase in service time for shared queues. After aggregating both direct and indirect effects, we find that pooling still leads to 6.86% longer service times in our focal supermarket. Finally, the direct effect and the indirect queue length effect function independently to each other.

Our research helps address how – and how much – queue configuration can affect human servers' service time in a supermarket setting, and makes three primary contributions to the literature.

First, our detailed data enables us to disentangle the direct effect from the indirect queue length effect. As a result, we are able to identify the significance and magnitude of both effects. Moreover, our investigation of the interaction between them reveals that they function relatively independently. There is a rich literature on the queue length effect, but none of these studies has explored how this effect may interact with queue configuration. Our study fills this gap.

Second, we find that the aggregation of both the direct and indirect effects of pooling on service times is positive. This provides empirical support for the claim that pooling queues can lead to longer service times (Rothkopf and Rech 1987). Considering that the checkout service is very simple and standard, the 10.7% difference in service time can be quite economically significant. Our research, based on supermarket data and studying richer mechanisms, complements that by Shunko et al. (2014), which is based on laboratory experiments. Our results also complement those obtained by Song et al. (2015) in that our human servers (i.e. cashiers) deal with non-discretionary tasks, do not have the flexibility to change the work process, and change their behavior based on different mechanisms.

Third, our study of the indirect queue length effect also confirms past research that concludes queue length impacts service time. Furthermore, we show that servers' service time is convex decreasing in the queue length in a supermarket, a non-discretionary work environment. This finding confirms the results by Lu et al. (2014).

Finally, our research also has important managerial implications. Without considering the indirect queue length effect on human servers, managers are likely to overstaff their systems. In addition, without considering the direct queue configuration effect on human servers, managers are likely to understaff their systems when pooling dedicated queues. Therefore, when considering pooling, managers must consider both the conventional operational benefits of pooling and, as our study suggests, service slowdown due to servers' behavioral changes. Our theoretical analysis in Section 6 incorporates both direct and indirect effects, and shows that if the social loafing effect is strong, pooling can hurt the system performance, particularly when the system load is either very high or very low.

2. Literature Review

Our research is related to studies of queueing system configurations. Conventional wisdom suggests that the pooling of queues usually leads to a reduction of the average waiting time (see, e.g., Eppen 1979; Kleinrock 1976; and Mandelbaum and Reiman 1998). That said, pooling may not be beneficial when the arrival streams being pooled have very different service time distributions and/or service level requirements (e.g., Whitt 1999). Rothkopf and Rech (1987) also conjecture that combining queues may lead to longer service times. One reason for such slowdown is the social loafing, or free riding, effect in shared queue – servers slow down to avoid being assigned more shared workload (Karau and Williams 1993; Krumm 2001). Considering a coordinating agency compensating two self-interested service providers to achieve a given expected waiting time, Gilbert and Weng (1998) show that a coordinating agency may prefer a separate queue configuration to a pooled one. Also, in their analysis of different dispatching policies in M/M/2 queues, Doroudi et al. (2011) highlight the importance of incorporating strategic server behavior in managing queueing systems. Do et al. (2015) use a theoretical model to study the impact of server behaviors, social loafing, and workload dependent speedup on the performance of dedicated queues and shared queues. They show that pooling may or may not lead to performance improvement.

There has been few empirical studies to verify the assumptions and results of these theoretical models, however. Studying case processing time in courts, Luskin and Luskin (1986)

argue that separating virtual queues of cases increases judges' individual accountability, reduces social loafing, and eventually leads to shorter case processing times; however, the estimated effect lacks statistical significance. Song et al. (2015) use an emergency department's patient-level data to show that a dedicated queueing system results in shorter throughput time and length of stay for patients when compared with a pooled queueing system. Their result cannot be explained by social loafing, however, because the hospital uses a so-called round robin routing policy, and the work allocated to a doctor does not dependent on her work speed. Rather, the more plausible explanation is that doctors have more ownership of patients in a dedicated queueing system and, thus, more actively manage the patient flow. In contrast, we study a supermarket setting, in which servers have no authority in managing the customer flow. Hence, any speedup effect in a dedicated queue can be attributed to the social loafing effect, instead of better management of the flow. Thus, our results in the simple, non-discretionary setting complement those in professional discretionary services, and help to establish the generality of the result that pooling leads to service slowdown. Shunko et al. (2014) show the slowdown effect of pooling queues in controlled lab experiments, but our results are derived from data that we collect from a supermarket, as we take advantage of the supermarket's unique design of a queueing system. In addition, none of the existing research, to the best of our knowledge, considers both a queue configuration's direct effect and its indirect queue length effect on service time; thus, the existing research fails to disentangle the two effects. Our research fills this gap by studying both effects, their interaction, and their aggregation.

Our research builds on and also complements the literature on queue length dependent service rate. On the theoretical side, Jackson (1963) is one of the early works that generalizes the exogeneity assumption and allows the service rate to depend on the queue length. Stidham Jr. and Weber (1989) and George and Harrison (2001) show that the optimal service rate should be nondecreasing in the queue length. Dong et al. (2013) study a queueing model that assumes a negative correlation between service rate and the congestion level of a queueing system. Also, Chan et al. (2014) allow for state-dependent service rates and analyze when speedup should be used, as well as this speedup's associated impacts. Finally,

Delasay et al. (2015) demonstrate that using models overlooking the state-dependent service rate may lead to wrong predictions of system performance and suboptimal staffing decisions.

In the supermarket environment that we study, customers waiting in queues clearly represent workload for servers, but this workload is virtual in the sense that these customers are waiting to be processed, but not actually being processed yet. In other service environments such as healthcare, servers simultaneously serve jobs in the system, so the workload is real. Accordingly, we will make a distinction in our study and use "queue" for the former and "workload" for the latter.

There is a recent stream of empirical research that investigates the impact of workload on server performance in various service environments. Both positive and negative correlations between workload and service time have been reported. Studying workload in a healthcare system, Kc and Terwiesch (2009) find that workers decrease their service time when there is a short-term workload increase, but a long-lasting high workload tends to increase service time. Kc and Terwiesch (2012) find a negative correlation between the workload and patients' length of stay in a cardiac intensive care unit. Also, Jaeker et al. (2012) find a positive correlation between real workload and patients' length of stay in a hospital. They further show that the anticipated high workload can be associated with either a longer or shorter length of stay, depending on the type of incoming workload. Using hourly sales as a performance measure instead of time, Tan and Netessine (2014) find that the correlation between service time and workload is positive when the overall workload is small, but negative when the overall workload is large. Armony et al. (2014), meanwhile, report a negative correlation between service rate and workload when an emergency department is crowded.

Our work is also closely related to empirical and experimental research on the impact of queue length on service time. For example, Edie (1954) documents that toll booth service time at the Lincoln Tunnel and the George Washington Bridge decreases with the traffic volume, because the server speeds up due to backed-up traffic, and drivers have more time to prepare payments before reaching the toll station. Studying a production system via a lab experiment, Schultz et al. (1998) show that workers speed up when they

are causing lines to be idle, and that the service time distribution depends on factors such as buffer size, co-workers' speed, and the amount of inventory in the system. Schultz et al. (1999) show that low-inventory systems improve productivity because of better feedback, group cohesiveness, and stronger task norms. Also, using data from a health care system, Batt and Terwiesch (2012) show that a patient queue affects service time through several mechanisms, including task reduction and load-induced slowdown; the net effect is an increase in service time when the system is crowded. Lu et al. (2014) study factors affecting workers' productivity in an IT service delivery system, and they find that workers' productivity is concave increasing with respect to the amount of waiting workload. Delasay et al. (2016) summarize mechanisms through which queue affects service time and their empirical evidence.

Recently, researchers have started to use video data to answer related questions. Using periodical queue information collected from video clips, Lu et al. (2013) empirically study how waiting in line affects customers' purchasing behavior. They find that waiting has a nonlinear impact on purchases, and customers focus mostly on the queue length, without fully adjusting for the speed that the line is moving. Jain et al. (2014) use video data and POS data to study how customers' in-store search and sales persons' assistance affect sales. In this paper, we also use video data, and combine it with the corresponding POS data. However, the research questions we study are different. In contrast to the focus on customer behavior in the two aforementioned papers, we focus on server behavior and study how queue configuration affects servers' service time.

3. Hypotheses Development

In this section, we develop testable predictions from theories about both the direct and indirect effects of queue configuration on service time, as well as their aggregation and possible interaction. We analyze the indirect effect of queue configuration on service time (via queue length) in §3.1, consider the direct effect of queue configuration in §3.2, and discuss the interaction and aggregation of the two effects in §3.3. In all, we develop hypotheses to test the various links depicted in Figure 1. All proofs are provided in the Appendices.

3.1. The Indirect Queue Length Effect

As clearly indicated in Figure 1, there are two effects that we must test: 1) how queue configuration affects queue length, and 2) how queue length affects service time. We start with the former.

Consider an M/M/2 queue with arrival rate 2λ and service rate μ, and an M/M/1 queue with arrival rate λ and service rate μ. We can show that the pooled M/M/2 queue has a smaller average queue length (see Proposition 3 in Appendix A). Hence, if everything else remains the same, the pooling of two identical M/M/1 queues leads to a shorter queue. Hence, queue configuration clearly has an impact on queue length.

This comparison assumes that different queue configurations do not affect customer arrivals, and also does not consider customers' queue joining behavior. However, in the supermarket where we collected our data, the two types of queues (dedicated queues and shared queues) are placed alternatively next to each other. Therefore, customers choose between different types of queues. Some customers are aware of the different queue configurations and their impact on expected waiting time, and incorporate this information when making queue joining decisions. In equilibrium, customers should be indifferent between joining the two types of queues. Because a shared queue has more servers and moves faster than a dedicated queue, we expect these informed customers to prefer a shared queue to a dedicated queue with the same length. Therefore, in equilibrium, the shared queues should be longer than the dedicated ones.

These two arguments result in different queue length comparison outcomes between the shared and dedicated queues. Therefore, we propose two competing hypotheses as follows, and let the data inform us which queue is shorter in our focal supermarket.

HYPOTHESIS 1. *(a) Shared queues are shorter than dedicated queues.*
(b) Shared queues are longer than dedicated queues.

In forming this hypothesis, we have made a conjecture about customers' queue joining behavior. To directly test such behavior would require additional data about each customer's choice set upon arriving at the checkout area. That focus is beyond the scope of this paper, but we believe that it deserves future research attention.

It should be noted that the test of Hypothesis 1 allows us to *indirectly* check whether customer queue joining behavior is present. If Hypothesis 1(a) is rejected and Hypothesis 1(b) is supported, then customers must be choosing queues strategically in some way.

Next, we study how queue length affects service time. In the literature on queue-dependent service rate, George and Harrison (2001) show that, if there is a pressure cost function that's weakly increasing in the queue length, then it is optimal for the server to use a state-dependent service rate that is non-decreasing in the queue length. This monotonicity has an intuitive appeal: when a queue is longer, the pressure on a server is higher, and any speedup effort by that server reduces waiting time for more customers; hence, the server is willing to work faster when the queue is longer.

There is very little empirical validation of this service rate policy, however, due to two practical obstacles. First, it is difficult to collect queue length data and match it with the service time data. Fortunately, the data set we collected contains POS transaction data as well as a matching set of video clips from which we can extract queue length information at any time (for details, see the data description in Section 4).

The second difficulty is that the existing research focuses on how *service rate* should optimally change with queue length, but service rate is never directly observed in practice. We can observe only the *service time* for each customer. Moreover, theoretically, the server should adjust her service rate whenever there is a new arrival (causing queues to increase). Hence, during the span of a customer's service time, the rate could change several times due to new arrivals. Following the literature and assuming exponential service times, each customer's total service time should be the sum of a random number of exponential random variables with different rates. Fortunately, we can overcome this obstacle in our study because we are able to develop a set of analytical results that translate the changes in service rate into the corresponding changes in each customer's service time (please see Appendix A for details), which we can then directly test.

Besides its effect on servers' behavior, a longer queue may also reduce the service time via its impact on customers. If a queue is long when a customer starts her service, it is most likely that the queue was already long when the customer joined the queue. As a result the focal customer then has more time to prepare her payment and to take items

from the shopping cart, both of which help speed up the service and reduce the service time.

Both factors imply that service time should be shorter when a queue is longer. Therefore, we propose the following hypothesis.

HYPOTHESIS 2. *The average service time decreases with the length of a queue at the start of service.*

Edie (1954) similarly shows that toll booth service time at a bridge is decreasing with the traffic volume, and attributes this to both queue pressure and the fact that drivers have more time to prepare payments. Because we are interested in the total effect of queue length on service time in this paper, we will not distinguish which of the two factors contribute more to a decrease in service time.

Although we hypothesize that servers work faster when queues are longer, the service rate cannot increase to infinity. Thus, we reasonably expect that the rate at which service rate increases will diminish as queue length continues to increase. In other words, service rate should be a concave increasing function of queue length. Although this has not been formally proved, numerical results in George and Harrison (2001) clearly indicate such a concave relationship. Assuming the service rate is concave in queue length, we can prove that the average service time should be a convex decreasing function of queue length (for details see Appendix A), which we can test using our data. We now formulate our next hypothesis:

HYPOTHESIS 3. *The average service time is a convex decreasing function of the queue length at the start of service.*

3.2. Queue Configuration's Direct Effect

Besides the indirect queue length effect, the queue configuration also directly affects a server's incentive and behavior. In the supermarket where we collected our data, servers are paid a fixed monthly salary, plus a bonus based on the number of transactions they complete within each month. We focus on the following two competing factors through which the queue configuration can affect servers' service rate:

1. *The social loafing effect*: A higher service rate requires a higher effort level, which results in a higher effort cost rate on servers. In addition, in a shared queue, working faster may also bring more work. Thus, servers working in shared queues have an incentive to slow down, when compared with those working in dedicated queues; in turn, this slowdown reduces not only the effort cost rate, but also the number of transactions allocated to servers.
2. *The competition effect*: Similarly, in a shared queue, the faster servers work, the more work they complete. This could have a stimulating effect on servers, however, as they get monthly bonuses based on the amount of work they complete. Therefore, more work means higher bonuses, and servers may work faster in shared queues due to competition for work.

It is worth pointing out that rational servers know that the sooner they finish their current transactions, the more likely that they are to get the next arrival to the shared queue. Therefore, social loafing and competition effects exist even when the queue length is zero.

As we just established, the social loafing effect and the competition effect work in opposite directions: for a faster server, the social loafing effect means more work assigned and a higher effort cost; conversely, the competition effect means a higher income. The optimal service rate will balance the trade-off between these two direct effects. If the bonus rate is low, then the competition effect is not strong enough to offset the social loafing effect, and servers should work slower in a shared queue than in a dedicated queue. In the extreme case when servers' compensation is unrelated to their completed work (e.g., when they are paid a fixed amount), the social loafing effect should dominate the competition effect. Conversely, if per transaction bonuses are sufficiently high, then the competition effect will dominate, and servers will work faster in the shared queue. For example, when servers share a common queue of potential buyers in a high-commission sales environment, they have the incentive to speed up a current customer, so they may compete with other servers to get a new customer.

In many settings, when servers change speed, there is a noticeable effect on the quality of their work. However, in the supermarket that we study, there are very high transaction accuracy requirements, and the servers almost always meet these standards. Thus,

there is little evidence of a quality-speed trade-off. Subsequently, we only focus on the impact of queue configuration on service speed. Because the social loafing and competition effects oppose each other, their total direct effect is ambiguous. We propose two competing hypotheses:

HYPOTHESIS 4. *(a) Servers work slower when working in shared queues, after we control for queue length.*

(b) Servers work faster when working in shared queues, after we control for queue length.

In order to separate the direct effect from the indirect queue length effect in our empirical tests, we compare servers' behavior in a dedicated queue with that in a shared queue, after we control for queue length.

3.3. The Interaction between and Aggregation of Direct Effect and Indirect Queue Length Effect

As both the direct effect and indirect queue length effect can exist simultaneously in a shared queue, we investigate the following questions in this section: What is their total effect on the service time? Furthermore, do they moderate each other? Does a longer queue affect the direct effect? If so, does the slowdown become more, or less, pronounced when the queue is longer?

As discussed in Section §3.2, the social loafing effect may cause servers to slow down when working in shared queues, because the faster a server works, the more work she will do. This effect could be stronger with a longer queue. When the queue is long, a server will clearly be assigned another customer from the queue as soon as she finishes serving the current customer; in contrast, with a shorter queue, the other server may finish the remaining work in the queue before the focal server finishes her current job. Based on this argument, the direct effect should be bigger when the queue is longer.

However, Morgeson and Humphrey (2008) suggest the opposite. They mention that social loafing is less likely to occur when the workload is higher, because everyone's contribution is needed and unique in such a case. In our setting, the indirect queue length effect could come from the pressure of keeping customers waiting. In a pooled queue, when the queue length is longer, both servers must work faster so as to reduce the queue length and

alleviate the pressure, which makes social loafing less likely to occur. On the other hand, when the workload is low, free riding does not affect performance much, and social loafing is therefore more likely to occur.

Based on these two different arguments, we propose the following competing hypotheses.

HYPOTHESIS 5. *(a) The direct effect is bigger when the queue is longer.*
(b) The direct effect is smaller when the queue is longer.

In Hypotheses 1-5, we test direct and indirect effects separately, as well as analyze how they affect each other. In practice, however, managers are most concerned with an overall aggregate effect on service time. Ultimately, managers want to know whether servers will slow down or speed up when queues are pooled, and theories alone do not answer this question definitively. To answer this question empirically, we propose the following two competing hypotheses.

HYPOTHESIS 6. *(a) At the focal supermarket, servers work more slowly when working in shared queues.*
(b) At the focal supermarket, servers work faster when working in shared queues.

4. Data

We used a primary data set that we collected from a supermarket in Shanghai, China to test our hypotheses. In this section, we explain how the data were collected, and then present key summary statistics of the final data set.

4.1. Data Collection Process

We obtained video records through closed-circuit television (CCTV) cameras from the supermarket's surveillance system covering the checkout area, and we use these records to observe customer arrivals and departures. The videos were recorded during all business hours. We also have staff scheduling records as well as Point-of-Sale (POS) transaction records containing details such as items purchased and their quantities, prices, and the payment methods. In 2014, we collected the data on two separate days: June 5th and June 7th. Each open POS terminal has one server to handle customer transactions on a first-come-first-served basis. When a server is serving an existing customer, newly arrived

Figure 2 Video Snapshot

customers entering the line need to wait until the departure of customers in front of them. Figure 2 shows a sample snapshot of the video from channel 3 recorded at 11:03:12 on June 7th, 2014. As can be seen, there is a queue of four customers, including the one being served, in front of POS 26.

In theory, customers may renege or jockey queues, but in the supermarket where we collected our data, queues leading to the POSs are separated by handrails. Thus, it is hard for a customer to leave or switch to another queue if there are other customers waiting behind her, particularly if she uses a shopping cart. Moreover, we rarely observe reneging and jockeying in our video observations. Therefore, we do not model these factors in our study.

The supermarket has two types of physical queue configurations: dedicated queues and shared queues. The former is a queue that leads to one POS; the latter has one shared queue leading to two parallel POSs. It is possible that only one of the two POSs is used in a shared queue at any time, in which case the shared queue effectively becomes a dedicated queue; accordingly, we will treat that queue as a dedicated queue in our empirical analysis. The two types of queues are placed in an alternating fashion, as illustrated in Figure 3. The management implemented such layouts mainly to better use the space in the checkout area. Servers are randomly assigned to POS terminals in each shift.

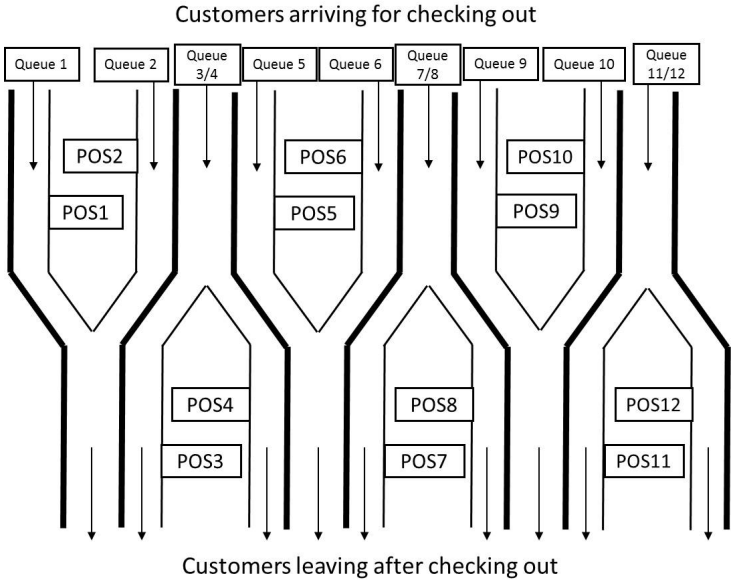

Figure 3 Layout of the POS

This unique design of the queueing system results in natural experiments ideal for comparing service time in the two types of queues and examining the impact of queue configuration on service time. In addition, as shown in Figure 3, POSs in dedicated queues are also placed in groups of two. If any peer effect (Mas and Moretti 2009) exists, then it should be present in both queue configurations. Moreover, as servers are randomly assigned in each shift, any peer effect that exists should be uncorrelated with the queue configuration and thus should not bias our estimation.

In a dedicated queue, customers wait and proceed to the server when that server becomes available. In a shared queue, customers wait and proceed to the first available server among the two. We define the arrival time of a customer as the time when the customer stops either behind the previous customer or in front of the server when the queue is empty. The departure time of a customer is defined as the time when the transaction is completed and the customer departs. If a customer arrives when the server is idle, then the service start time is the arrival time. If a customer arrives when the server is busy, then her service starts when the server finishes serving the customer ahead of her, which is the same as the departure time of the present customer ahead of her. The service time is defined as the

difference between a customer's departure time and the service start time. Queue length is defined as the number of customers in a queue, excluding the customer(s) being served.

Research assistants watched videos to collect raw data of the arrival time and departure time of each customer, as well as the presence time of each server at the POS. They were given a timer program in Excel that has buttons corresponding to the data to be collected for each channel. To minimize errors, each channel was assigned to two research assistants for data recording, and the two sets of raw data were compared and cross-checked. Any discrepancy was then inspected by a third assistant, who would watch the related video to make a final determination. In addition, the POS records have a time stamp for each transaction. We also compared the times recorded from the video with those from POS transaction records. Any unusual differences were inspected and reconciled.

The POS transaction records do not contain information about queue length. Fortunately, we can deduce the queue length at any time from all the customers' arrival and departure times. The difference between the total number of customer arrivals and the total number of customer departures by a certain time is the queue length at that time. For example, at 9:00am, our record shows four customer arrivals since the opening of a POS, but only two customer departures. Therefore, the other two customers, including the customer who is being served at that moment, are still in the system. Therefore, our queue length is 1. Some customers arrive by groups. For example, a couple may go shopping together. Customers within one group typically have similar departure times, which enables us to identify groups. For our analysis, we define the queue length as the number of waiting individuals. We have also repeated the empirical analysis by using the number of waiting groups instead of individuals; when we do so, the directional results remain the same.

Using the departure times observed from the video and the transaction times from the POS transaction records, we can match customers observed in the video with transactions in the POS transaction records. Therefore, for each customer transaction, we have the corresponding service time, the queue length at the start of the transaction, and purchase details such as the number and type of items purchased, the total amount purchased, the server ID, and the payment method. We drop a few outliers that are unusually large.

Table 1 Summary Statistics of Key Variables

Variable	All Transactions	Dedicated Queue	Pooled Queue	Difference Pooled - Dedicated	P-value
Service time	77.32	76.43	80.06	3.63	0.04
Queue length	1.39	1.31	1.62	0.31	0.00
Number of normal items	6.09	6.07	6.15	0.08	0.82
Number of grocery	1.11	1.11	1.10	0.00	0.97
Total value	69.51	70.19	67.41	-2.78	0.23
Total SKU	4.95	5.00	4.80	-0.20	0.19

4.2. Summary Statistics

The resulting data set contains 4,305 transactions. Table 1 presents summary statistics of some key variables. The second column shows variable means across all transactions. The average service time for a transaction is about 77.32 seconds. In this table, and throughout the rest of the paper, "queue length" stands for queue length at the beginning of each transaction. Its average is 1.39 across all transactions. Products are divided into two groups. Grocery refers to goods that servers must weigh to determine their amount, whereas normal items refer to goods that only require bar code scanning. The average number of normal items in a transaction is about 6.09, and the average number of grocery is about 1.11. The average total value of each transaction is 69.51 RMB. The number of stock keeping units (SKUs), which indicates how many distinct items were sold in a transaction, is 4.95 on average. As mentioned earlier, there are two types of queues in the supermarket: dedicated queues and shared queues. The third and fourth columns of Table 1 present the mean of key variables in dedicated queues and shared queues, respectively. The fifth column shows the mean differences between shared queues and dedicated queues. We also conduct unpaired T-tests between the two types of queues, and report the P-values in the last column. The transaction characteristics (numbers of normal items, number of grocery, total value, and total SKU) are not statistically different between the two types of queues, indicating that transactions in the two types of queues are similar. However, both the service time and queue length are significantly longer in shared queues than in dedicated queues.

Table 2 shows the correlation matrix of the key variables. The maximum absolute value of correlation between the queue length and any other variable is 0.07, which clearly

indicates that queue length is not correlated with any transaction characteristics variable. As expected, some transaction characteristics have positive correlations. For example, the total value and the total SKU have a correlation of 0.76, because the more items customers buy, the more they spend. The transaction characteristics are control variables in our empirical study, so correlations among themselves should not have significant effects on estimating the impacts of our main explanatory variables on service time.

Table 2 Correlations of Key Variables

	(1)	(2)	(3)	(4)	(5)
(1) Number of normal items	1.00				
(2) Number of grocery	0.02	1.00			
(3) Total value	0.48	0.25	1.00		
(4) Total SKU	0.55	0.44	0.76	1.00	
(5) Queue length	0.00	-0.07	-0.05	-0.07	1.00

Figure 4 shows the distribution of service time, which clearly does not follow any exponential distribution. As shown in the histogram of log(service time) in Figure 5, the distribution of the natural log of the service time is very close to a normal distribution. The lognormal Q-Q plot of service time, as presented in Figure 6, shows almost a straight line and further supports that a lognormal distribution is a good representation of service time. This finding in our supermarket checkout setting confirms and extends the research in call centers that finds the distribution of call durations to be approximately lognormal (e.g., Bolotin 2013; Brown et al. 2005).

In the next section, we use the resulting data set to empirically test Hypotheses 1- 6.

5. Empirical Models and Results

In our data, each observation corresponds to one transaction. As discussed in Section 4.2, the service time follows approximately a lognormal distribution and has a long right tail; therefore we follow the literature (e.g., Kc and Terwiesch 2009; Song et al. 2015) and use natural log of service time as our dependent variable in the empirical models.

To control transactions, servers, location, and time heterogeneities, we include transaction characteristics, payment method dummies, server dummies, POS dummies, and

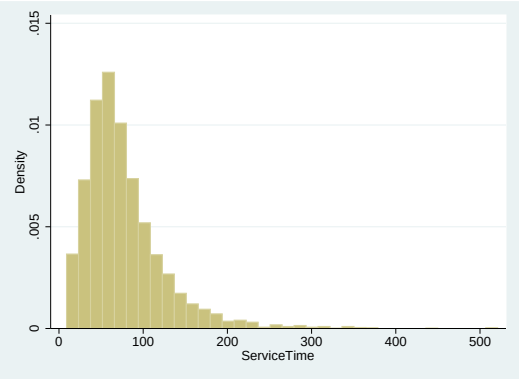

Figure 4 Service Time Distribution

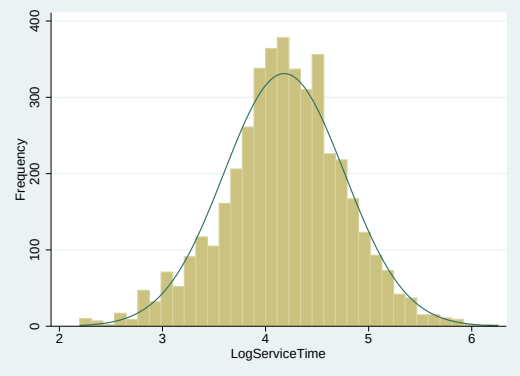

Figure 5 Fitting Log of Service Time with Normal Distribution

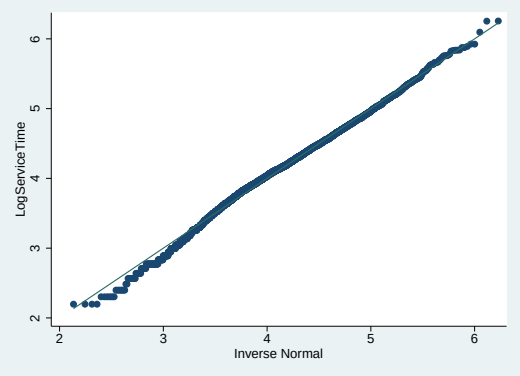

Figure 6 Normal Quantile Plot for Log of Service Time

day-hour dummies as control variables in our empirical models. In creating the day-hour dummy variables, we treat the same hour on different days as different time periods. We also include the number of common SKUs in both normal items and grocery items between the focal transaction and the previous transaction to control for the commonality between consecutive transactions – and the possible impact of servers' short-term familiarity with the items on their service time. In addition, we include the total number of open POS stations in the whole store, so we may control for the store-wide social loafing effect that exists among all queues. Table 3 provides a complete list summarizing all the control variables.

Service time can be viewed as the amount of work in a transaction divided by the server's speed. Transaction characteristics, including $NumberGrocery$, $NumberItems$,

Table 3 Control Variables

Variables	Definition
Transaction characteristics	
log(NumberItems)	Log of the number of normal items in a transaction
log(NumberGrocery)	Log of the number of grocery items in a transaction
log(TotalValue)	Log of the total value of a transaction
log(TotalSKU)	Log of the number of SKUs in a transaction
Payment method dummies	Whether the transaction involved payment using a particular type of payment method
Server dummies	Whether the transaction was completed by a particular server
POS station dummies	Whether the transaction was completed in a particular POS station
Hour-day dummies	Whether the transaction started in a particular hour on a particular day
Common normal items	The number of common normal items (by SKU) between the previous transaction and the focal transaction
Common grocery items	The number of common grocery items (by SKU) between the previous transaction and the focal transaction
Number of open POS	The number of open POS stations in the store within the hour

$TotalValue$, and $TotalSKU$, directly affect the work amount in a transaction, but not the server speed. Therefore, the service time should change linearly to these variables. On the other hand, queue configuration and queue length may affect the server speed; hence, their impact on service time should be proportional to the work amount of each transaction. That is, the absolute value of queue configuration effect or queue length effect should be larger when the transaction amount is higher. Moreover, since we know the transaction characteristics variables are highly right-skewed, we use their natural logs instead of their levels as control variables. Because the dependent variable is the log of service time, the relationship between service time and these transaction characteristics variables is linear. For robustness checks, we also ran regressions with levels of transaction characteristics as control variables, and report the results in Appendix C. Estimation results show that regressions using natural logs of transaction characteristics have higher R^2, and thus better fit, than those using levels of transaction characteristics.

Considering the supermarket setting where our data were collected, we do not think endogeneity is a big concern with respect to our empirical estimations. There are two factors that determine service time: work amount and server speed; the work amount depends on transaction characteristics such as the basket size. Because dedicated queues and shared queues are placed in an alternating fashion as demonstrated in Figure 3, we do not expect that transactions in dedicated queues are systematically different from those in shared queues. The statistics and T-test results in Table 1 also confirm the similarity of transactions in both types of queues. The other factor, server speed, is dependent on the server. In the focal supermarket, servers are randomly allocated to different POSs, and we also include server and POS dummies as control variables. Therefore, endogeneity due to omitted server-related or POS-related variables should not be an issue.

When customers reach the checkout area, they naturally select the queue that they believe will ensure maximum expediency. Thus, one may suspect such queue selection behavior to cause endogeneity issues. However, although we believe customers' queue selection decision may be correlated with queue length or queue configuration, this decision should be uncorrelated with any customer-specific factor that may affect service time. We would be concerned if customers with different transactions demonstrated different queue selection behavior. That said, we would not anticipate such phenomena in our setting, because all customers make queue selection decisions to minimize their expected waiting, and should make similar decisions when facing the same choices. Therefore, customers' queue selection behavior is unlikely to correlate with unobserved factors affecting the service time, and should not cause endogeneity issues in our estimations.

The errors within transactions over the same server/time period/POS may be correlated, which may not be fully controlled for by the fixed effects. Therefore, when we report results, we use cluster-robust errors, in which transactions conducted by the same server in the same hour at the same POS station are treated as being in one cluster.

5.1. The Indirect Queue Length Effect

Hypothesis 1 predicts whether shared queues are longer than dedicated queues. To test this hypothesis, we use the queue length when transaction i starts as the dependent variable, and the dummy variable $SingleServerQueue_i$, which indicates that the transaction

is completed in a dedicated queue instead of a shared queue, as the main explanatory variable. We also include the hour-day dummies to control for any time heterogeneities. The resulting empirical model is

$$QueueLength_i = \beta_0 + \beta_1 SingleServerQueue_i + \vec{\gamma}\overrightarrow{HourDayDummies}_i + \epsilon_i. \quad (1)$$

The queue length may be autocorrelated across subsequent transactions in the same queue, which means that the error terms in Equation 1 may be correlated with each other. We used the generalized least squares (GLS) method to incorporate such autocorrelations. We specify the data as a panel data with the POS stations as panels, and allow the error terms to be autocorrelated within each panel. We then use the panel-specific AR1 autocorrelation structure to allow different queues to have different autocorrelation patterns. The estimated coefficient of $SingleServerQueue_i$ is -0.216, which is statistically significant at the 5% significance level, and also economically significant since the overall average queue length across all transactions is 1.39, as reported in Table 1. This result indicates that dedicated queues are shorter than shared queues on average and supports Hypothesis 1 (b). For brevity, we do not report the coefficients of the hour-day dummies. This result also indirectly supports our conjecture that, given a fixed queue length, customers are more likely to choose a shared queue than a dedicated queue.

Hypothesis 2 states that the average service time is a decreasing function of the queue length. Our main explanatory variable is the queue length at the time when a transaction starts. We use transaction characteristics and other control variables as listed in Table 3, and obtain this resulting model:

$$log(ServiceTime_i) = \beta_0 + \beta_1 QueueLength_i + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i. \quad (2)$$

To test Hypothesis 3, which suggests that queue length has a marginally diminishing impact on service time, we allow for a nonlinear relationship between the dependent variable and the main independent variable by including $QueueLength_i^2$ in addition to $QueueLength_i$[1]:

$$log(ServiceTime_i) = \beta_0 + \beta_1 QueueLength_i + \beta_2 QueueLength_i^2 + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i. \quad (3)$$

Table 4 Impact of Queue Length

VARIABLES	(2) LogServiceTime	(3) LogServiceTime
QueueLength	-0.0480**	-0.0938**
	(0.00643)	(0.0132)
QueueLength^2		0.00960**
		(0.00218)
Log(NumberItems)	0.103**	0.103**
	(0.0158)	(0.0157)
Log(NumberGrocery)	0.115**	0.112**
	(0.0151)	(0.0152)
Log(TotalValue)	0.103**	0.102**
	(0.0120)	(0.0121)
Log(TotalSKU)	0.115**	0.115**
	(0.0193)	(0.0190)
Control Variables	Included	Included
Observations	3,245	3,245
R^2	0.476	0.479

Robust standard errors in parentheses
** p<0.01, * p<0.05

To isolate the impact of queue length on service time, we used data from only one queue configuration, and chose to focus on the dedicated queues in this paper[2]. Table 4 summarizes the results when we estimate the empirical models (2) and (3). For simplicity, we only report the queue length effect results and the coefficients of transaction characteristics, and omit coefficients of other control variables. We find that the semi-elasticity of queue length on service time is -0.048 in Model (2), which is statistically significant at the 1% significance level. This result provides support to our hypothesis that service time is decreasing in the queue length.

[1] We have also tested specifications other than quadratic and found our main results are robust. The estimation results of other specifications are available from the authors upon request.

[2] We have also run regressions using data from shared queues. The results are similar and only reported in Appendix B for the sake of brevity.

However, this speedup effect is diminishing as queue length increases, as evidenced by results in Model (3). Allowing a quadratic relationship between queue length and the logged service time, the coefficients of $QueueLength_i^2$ and $QueueLength_i$ are 0.0096 and -0.0938, respectively. The estimation results imply that when the queue length increases from zero to one, the service time decreases by about $0.0938 - 0.0096 = 8.42\%$;[3] when the queue length increases from one to two, the decrease in service time is about $0.0938 * 2 - 0.0096 * 2^2 - 8.42\% = 6.5\%$; when the queue length increases from two to three, the decrease in service time is only about $0.0938 * 3 - 0.0096 * 3^2 - (0.0938 * 2 - 0.0096 * 2^2) = 4.58\%$; etc. In addition, adding a quadratic term increases the adjusted R^2 from 0.467 to 0.470, which suggests that Model (3) is a better model; as a result, we include both $QueueLength_i$ and $QueueLength_i^2$ in the related estimation models for the remainder of this paper.

In addition, the coefficients of transaction characteristics, such as the logarithm of number of items purchased in a transaction, are all positive and statistically significant, confirming the intuition that a larger transaction takes more time to serve.

5.2. Queue Configuration's Direct Effect on Service Time and Its Interaction with the Indirect Queue Length Effect

We now use the supermarket data to test Hypotheses 4(a)-4(b) regarding the direct impact of queue configuration on service times, and also to test Hypotheses 5(a)-5(b) regarding the interaction between the direct and indirect effects.

In discussing the social loafing effect, we assume that a dedicated queue is free of social loafing effect for the sake of simplicity. In practice, when customers arrive at the checkout area, they need to decide which queue to join. Exploring how customers select queues remains an interesting question, but beyond this paper's focus. With respect to our study, such queue selection behavior creates a social loafing effect across all servers. Even in dedicated queues, a slow server will have customers accumulate in the queue, which reduces the likelihood that a newly arrived customer will choose to join that particular queue. Therefore, in terms of customer queue selection behavior, the social loafing effect exists

[3] We are following the convention of using first-order approximation to calculate the percentage changes here.

across all servers, regardless of queue configuration. As stated earlier, we use a variable *Number of open POS* (see Table 3) to control for such store-wide social loafing effect.

However, there is an additional incentive for the two servers sharing the same pooled queue to slow down, and that is what we focus on in this paper. In a dedicated queue, once a customer enters a server's queue, the server has to serve this customer. In a shared queue, however, even after a customer joins the queue, it remains unclear which of the two servers will serve the customer. The slower a server works, the smaller portion of the shared queue will come to that server. Therefore, the social loafing effect within each shared queue that we study is in addition to the aforementioned store-wide social loafing effect.

The portion of a shared queue served by a server clearly depends on that server's service speed. Whether a server prefers to speed up or slow down depends on the level of transaction-based bonus. If the bonus is small, the social loafing effect should dominate the competition effect, and the server would prefer to slow down and free ride on the other server; in that case, Hypothesis 4(a) should hold. When the bonus is sufficiently large, servers should prefer to get more work, and the competition effect should dominate. Therefore, servers should speed up, and Hypothesis 4(b) should hold. To test which hypothesis holds for the focal supermarket, we use $SingleServerQueue_i$ as our main explanatory variable.[4] This binary variable equals one if the transaction occurs at a POS with a dedicated queue, instead of a POS sharing a queue with another POS.

To control for the indirect queue length effect, we include $QueueLength_i$ and $QueueLength_i^2$ as control variables, based on the result in Section 5.1 that the queue length effect on service time is quadratic. In addition, Hypothesis 5(a) predicts that the direct effect, whether positive or negative, should be amplified as the queue length increases, whereas Hypothesis 5(b) predicts the opposite. Therefore, besides $QueueLength_i$ and $QueueLength_i^2$, we also include their interactions with the main explanatory variable $SingleServerQueue_i$, $SingleServerQueue_i * QueueLength_i$ and $SingleServerQueue_i *$

[4] We have also conducted a robustness check and confirm that our results are still robust even if we allow for server heterogeneity in the direct effect. Detailed estimation results of this robustness test are available from the authors upon request.

$QueueLength_i^2$, in the empirical model. All the control variables in Table 3 are still included. The resulting empirical model is:

$$\begin{aligned} log(ServiceTime_i) =& \beta_0 + \beta_1 SingleServerQueue_i + \beta_2 QueueLength_i + \beta_3 QueueLength_i^2 \\ &+ \beta_4 SingleServerQueue_i * QueueLength_i \\ &+ \beta_5 SingleServerQueue_i * QueueLength_i^2 + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i. \end{aligned} \tag{4}$$

The estimation results of Model (4) are reported in the second column of Table 5. For the sake of brevity, we omit the coefficients of the control variables. The coefficient of $SingleServerQueue_i$ is statistically significant and estimated to be -0.107 in Model (4), which means that, after we control for queue length, the service time is approximately 10.7% shorter if the transaction occurs in a dedicated queue than in a shared queue, which empirically supports Hypothesis 4(a) and rejects the competing Hypothesis 4(b).

Table 5 The Total Direct Effect, its Interaction and Aggregation with the Queue Length Effect

	(4)	(5)
VARIABLES	LogServiceTime	LogServiceTime
SingleServerQueue	-0.107**	-0.0686*
	(0.0327)	(0.0269)
QueueLength	-0.113**	
	(0.0185)	
QueueLength^2	0.00930**	
	(0.00285)	
SingleServerQueue	0.0206	
*QueueLength	(0.0219)	
SingleServerQueue	-8.33e-06	
*QueueLength^2	(0.00344)	
Control Variables	Included	Included
Observations	4,305	4,305
R^2	0.474	0.452

Robust standard errors in parentheses
** p<0.01, * p<0.05

In addition, the coefficients of interaction variables, $SingleServerQueue * QueueLength$ and $SingleServerQueue * QueueLength^2$, are statistically insignificant, which means that the direct effect is not affected by the queue length, and the server becomes uniformly slower when working in a pooling queue. In other words, the queue configuration's direct effect and indirect queue length effect function independently of each other. Therefore, we reject both Hypotheses 5(a) and 5(b). This result also suggests that the impact of queue length on service speed is similar in dedicated queues and shared queues.

5.3. The Aggregate Effect of Pooling

Though the queue configuration's direct and direct effects function independently, they are both present simultaneously. In practice, what matters for managerial decision making is the total queue configuration effect on service time. Because the two effects can work in opposite directions, it is interesting to know whether the overall service time is longer or shorter in a shared queue when compared to that in a dedicated queue. The general answer to this question depends on specific system parameters, such as overall customer traffic, what other queues are present, and how consumers make their queue-joining decisions. In what follows, we use our data to compare the service time for servers working in dedicated queues with those in shared queues, while we control for transaction characteristics and possible heterogeneities. We intentionally exclude all variables related to queue length, so we may aggregate both the direct effect and the indirect queue length effect. The resulting empirical model is as follows:

$$log(ServiceTime_i) = \beta_0 + \beta_1 SingleServerQueue_i + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i. \tag{5}$$

The estimation results are presented in the third column of Table 5. The coefficient of $SingleServerQueue$ (-0.0686) is negative and statistically significant, which indicates that servers are slower when working in shared queues; this finding supports Hypothesis 6(a) and rejects Hypothesis 6(b). In our setting, even though shared queues are longer and pressure servers to speed up, the social loafing effect is stronger and dominates. However, this result depends on how pooling affects the queue length, and should not be generalized to other service systems without a similar empirical test that accounts for application-specific factors.

5.4. Robustness Tests

In this section, we conduct several robustness tests.

5.4.1. Impact of Complexity of Transactions in Queue on Service Time In the supermarket, besides the queue length, the amount of products to be checked out in the queue is also partially visible to the server. Therefore, it is possible that a server also uses such information about the complexity of transactions in the queue to adjust her service rate. We now consider whether the complexity of transactions in the queue, besides the queue length, affects servers' service time. To do so, we add two additional independent variables, $WaitingItems_i$ and $WaitingGrocery_i$, which record the number of normal items and grocery items in the queue, to Models (2) and (3). The resulting models are

$$\begin{aligned} log(ServiceTime_i) =& \beta_0 + \beta_1 QueueLength_i + \beta_2 WaitingItems_i \\ &+ \beta_3 WaitingGrocery_i + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i, \end{aligned} \tag{6}$$

and

$$\begin{aligned} log(ServiceTime_i) =& \beta_0 + \beta_1 QueueLength_i + \beta_2 QueueLength_i^2 + \beta_3 WaitingItems_i \\ &+ \beta_4 WaitingGrocery_i + \vec{\gamma} \cdot \overrightarrow{Controls}_i + \epsilon_i. \end{aligned} \tag{7}$$

The estimation results of Models (6) and (7) are shown in Table 6 (again, we omit coefficients of all control variables). The coefficients of $WaitingItems_i$ and $WaitingGrocery_i$ are statistically insignificant, which means that the complexity of transactions in the queue does not affect servers' working speed after we control for queue length; instead, what matters most is the number of waiting customers.

5.4.2. Impact of Queue Configuration on Customers' Behavior In supermarkets, customers play an important role in checkout service times. Then does the queue configuration directly affect customers' behavior? We answer this question by analyzing whether there is any difference in the customer's basket, which is under the control of customers, between the two types of queue configurations.

We use the natural log of variables related to customers' baskets (*log(NumberItems)*, *log(NumberGrocery)*, *log(TotalValue)*, *log(TotalSKU)*) and the percentage of grocery items,

Table 6 Impact of Complexity of Transactions in Queue on Service Time

VARIABLES	(6) LogServiceTime	(7) LogServiceTime
QueueLength	-0.0497**	-0.0951**
	(0.00673)	(0.0135)
QueueLength^2		0.00957**
		(0.00219)
WaitingItems	8.85e-05	8.14e-05
	(0.000116)	(0.000119)
WaitingGrocery	0.00213	0.00189
	(0.00280)	(0.00273)
Control Variables	Included	Included
Observations	3,245	3,245
R^2	0.476	0.479

Robust standard errors in parentheses
** p<0.01, * p<0.05

defined as the number of grocery item divided by the total number of all items, as the dependent variables respectively, and use the dummy *SingleServerQueue* as our main explanatory variable. In order to control for time and location heterogeneities, we include the day-hour dummies and POS dummies as control variables. The estimation results show that the coefficient of *SingleServerQueue* in all five regressions is always statistically insignificant, which means that transactions in the two types of queues do not exhibit significant difference in basket size, number of items, type of items, and so forth. Such results imply that the difference in service time between dedicated queues and shared queues cannot be explained by changes in customers' baskets, which is under the control of customers, and indirectly support that service slowdown in shared queues is mainly driven by the servers' behavior.

5.4.3. Subsample Analysis In the supermarket, servers are randomly allocated to POS stations. Because our data set covers two days, for some servers, we only observe them working in one type of queueing system. For example, some servers worked in dedicated queues on both days. Excluding transactions completed by servers who worked in only one

Table 7 The Direct Effect, its Interaction and Aggregation with the Queue Length Effect: Subsample Analysis

VARIABLES	(4) LogServiceTime	(5) LogServiceTime
SingleServerQueue	-0.108**	-0.0726*
	(0.0322)	(0.0291)
QueueLength	-0.109**	
	(0.0190)	
QueueLength^2	0.00888**	
	(0.00288)	
SingleServerQueue *QueueLength	0.0304 (0.0223)	
SingleServerQueue *QueueLength^2	-0.00184 (0.00348)	
Control Variables	Included	Included
Observations	2,780	2,780
R^2	0.487	0.463

Robust standard errors in parentheses
** p<0.01, * p<0.05

type of queueing system on the two days, we have a subsample with 2,780 observations. To further address the server heterogeneity concern, we also estimate Models (4) and (5) by using the subsample of transactions completed by servers who were observed working in both types of queue configurations, and we report our estimation results in Table 7. All our results are robust. For example, the coefficient of $SingleServerQueue$ in Model (4) is negative, which indicates that the direct effect of pooling results in longer service times and supports Hypothesis 4(a).

5.4.4. Using Levels of Transaction Characteristics as Control Variables We have tried using levels of transaction characteristics variables $NumberGrocery_i$, $NumberItems_i$, $TotalValue_i$, and $TotalSKU_i$ as control variables for Models (2) to (5), rather than their natural logs. When we do so, all results still hold. We have also used the service time itself instead of its log value as the dependent variable, with levels of trans-

action characteristics as control variables, to estimate Models (2) to (5). Again, all results remain valid. For brevity's sake, we report the detailed estimation results in Appendix C.

6. Managerial Implications

In most cases, pooling similar queues helps to reduce waiting. In this paper, we empirically identified two server behaviors that may change this view. First, the competition and social loafing effects suggest that servers may work faster or more slowly in a pooled queue, depending on the incentive. Second, the queue length effect means that pooling can further change servers' speed in a pooled queue via its impact on queue length. In this section, we incorporate both of the servers' behavioral factors into simple M/M/-type queues to examine the conditions under which pooling is beneficial. We believe that the results in this section can be used in more complex queueing systems in which similar server behaviors are important factors, and the insights generated in this section can help managers compare different queue configurations.

Consider two identical and independent M/M/1 queues with queue length dependent service rates. Let λ be the Poisson arrival rate, and let $\mu_q, q = 1, 2, \ldots$ denote the exponential service rate when the queue length is q. After we pool the two queues, the arrivals continue to follow a Poisson process with rate 2λ. We further assume that service rates in the shared queue still depend on queue length: each server works at an exponential rate $\mu'_q = d \cdot \mu_q$ when the queue length is q. We assume $d = \mu'_q/\mu_q$ for all q, based on our finding that the direct effect functions independently of queue length.

Further, the parameter d captures the direct effect. When $d > 1$, the competition effect dominates, so servers actually work faster when facing a shared queue. When $d < 1$, the social loafing effect dominates, and servers work slower when facing a shared queue. The case $d > 1$ is quite obvious since the competition effect further enhances the pooling benefit. For the rest of this section, we will focus on the case of $d < 1$, which is managerially more interesting and important.

Our analysis proceeds in two steps. First we assume that there is no queue length effect, and only focus on the direct effect represented by d. This allows us to isolate the direct effect and generate appropriate insights. Thereafter, we add the queue length effect to see the aggregate effect.

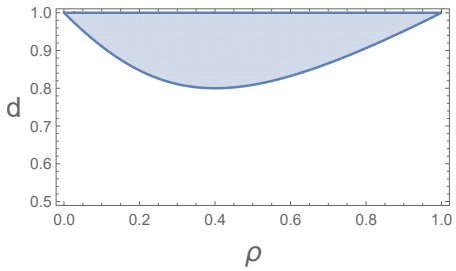

Figure 7 Region where Pooling Decreases W_s

PROPOSITION 1. *Assume $\mu_q = \mu$ for all q. Let $\rho = \lambda/\mu$ and denote by W_s the average time in the system. Then there exits $\bar{d}(\rho) = \frac{1-\rho+\sqrt{(1-\rho)^2+4\rho^2}}{2}$ such that,*

1. *When the social loafing effect is mild, i.e., $d > \bar{d}(\rho)$, pooling reduces W_s;*
2. *When the social loafing effect is strong, i.e., $d < \bar{d}(\rho)$, pooling increases W_s.*

For the ease of exposition, we will define "standard pooling benefit" as the reduction in W_s due to pooling when $d = 1$ and $\mu_q = \mu$ for all q.

Proposition 1 makes intuitive sense. The standard pooling benefit reduces waiting time in the system, but the social loafing effect slows down service and increases waiting time. Hence, when the social loafing effect is strong, the service slowdown can more than offset the standard pooling benefit; in such a case, managers should not pool queues together. Conversely, when the social loafing effect is weak, the server slowdown does not offset the standard pooling benefit, so pooling queues remains beneficial.

Clearly from Proposition 1, the threshold on the social loafing factor, $\bar{d}(\rho)$, is a function of system load, ρ. The next corollary shows how the threshold $\bar{d}(\rho)$ changes with respect to ρ:

COROLLARY 1. *$\bar{d}(\rho)$ is decreasing in ρ if and only if $\rho < 2/5$.*

Figure 7 shows the region where pooling is beneficial, even after considering the direct effect. For a fixed direct effect d, pooling is more likely to be beneficial when the load of the system is intermediate, but more likely to hurt the performance when the load is high or low. When the load is low, the time in the system is mostly determined by the service time itself; slowdown due to social loafing, then, can increase W_s because the average service time is longer after pooling. When the load is high, the waiting time is very sensitive to a

further increase of the load, then the social loafing effect increases the load and, thus, also increases W_s. Therefore, in service environments where the social loafing effect is present, it is unwise to pool queues together when the load is either very low or very high. The following numerical example, which uses parameters calibrated from our supermarket data, serves to illustrate this point.

Numerical Example: In the supermarket that we study, the estimated direct effect of pooling is an increase of service time by about 10.7%, which corresponds to $d = 0.893$. *Solving* $\bar{d}(\rho) = 0.893$, *we have two roots 0.124 and 0.769. Therefore, considering the direct effect only, using shared queues can reduce the average waiting time when the load factor (based on service rate in dedicated queues) is between 0.124 and 0.769.*

So far, we have focused only on the direct effect'. Next, we incorporate queue length dependent service rates into the model above. We do so to reflect our empirical finding that service time is increasing in queue length. We assume that μ_q is an non-decreasing function of q. Although our empirical results also reveal that $1/\mu_q$ decreases in a convex way, our analysis below is more general and does not assume as much. We allow μ_q to be any non-decreasing function of q.

Proposition 2 below extends Proposition 1 by showing that pooling can lead to a larger W_s if the social loafing effect is strong, even after the indirect queue length effect is incorporated:

PROPOSITION 2. *Let* μ_q *be non-increasing in* q. *There exists a* $\hat{d}$ *such that pooling increases* W_s *if and only if* $d < \hat{d}$.

Thus, managers should not pool queues together if the social loafing effect is strong.

7. Concluding Remarks

In this paper, we study the impact of queue configuration on the service time of human servers in a supermarket checkout setting, by comparing queues dedicated to specific servers with queues that are shared by two servers. The queue configuration could have both direct and indirect effects on the service time. Directly, the social loafing theory predicts that servers slow down when working in shared queues; however, servers working in

shared queues may also speed up in order to compete for transaction-based bonuses. Indirectly, pooling may affect queue length, for a longer queue puts pressure on servers to work faster. We investigate these effects and test our associated hypotheses using a data set collected from a supermarket's checkout process. We find that the average service time is convex decreasing in queue length in both dedicated queues and shared queues. Hence, our other finding – that shared queues are longer than dedicated queues – means that pooling has an indirect negative effect on service time, through its impact on queue length. We also find that the social loafing effect dominates the competition effect, and the average service time in shared queues is approximately 10.7% longer than that in dedicated queues, after we control for the queue length. In addition, we find that the direct effect and indirect queue length effect function independently from each other. Finally, the aggregate impact of pooling, including both direct and indirect effects, is a 6.86% increase in the average service time. These results are robust to alternative model specifications.

We then incorporate these empirical findings into a standard queueing model to analyze the impact of human behavioral factors on queueing performance. Our results indicate that the pooling benefit is not only smaller than that suggested by a model that ignores its effects on human servers, but can also even be negative in certain cases. When the social loafing effect is strong, pooling can hurt the system performance, particularly when the system load is either very high or very low.

Our research certainly has its limitations that future research can address. First, as behavioral effects are complicated and can be very context specific, repeating our study in different service settings to observer whether conclusions are similar would prove worthwhile. Second, there are other mechanisms and behavioral effects in queueing systems that are worth studying, including whether performance is below the subgoal (Deo et al. 2014), deadline effect (Deo et al. 2014), and end of shift effect (Chan et al. 2014). Third, customers' queue joining behavior is an interesting topic that deserves further research. Finally, in service industries with high personal contact, customers' behaviors also affect service performance (Feldman et al. 2014), which is also a deserving topic for future study.

Acknowledgments

The authors thank Serguei Netessine (the department editor), the associate editor, and three anonymous referees for their comments and help through the review process. The authors are also grateful to Kenneth Schultz, Armann Ingolfsson, Masha Shunko, and Tom Tan for their constructive suggestions. This research was supported by a seed fund for basic research at the University of Hong Kong.

References

Armony, M, S Israelit, A Mandelbaum, Y Marmor, Y Tseytlin, G Yom-Tov. 2014. Patient flow in hospitals: a data-based queueing-science perspective. The Technion, Haifa, Israel.

Batt, Robert J, Christian Terwiesch. 2012. Doctors under load: An empirical study of state-dependent service times in emergency care. *Working Paper* .

Bendoly, Elliot, Rachel Croson, Paulo Goncalves, Kenneth Schultz. 2010. Bodies of knowledge for research in behavioral operations. *Production and Operations Management* **19**(4) 434–452.

Bolotin, V. 2013. Telephone circuit holding time distributions. *Proceedings of the ITC*, vol. 14. 125–134.

Brockmeyer, E, HL Halstrm, Arne Jensen, Agner Krarup Erlang. 1948. The life and works of ak erlang. .

Brown, Lawrence, Noah Gans, Avishai Mandelbaum, Anat Sakov, Haipeng Shen, Sergey Zeltyn, Linda Zhao. 2005. Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American statistical association* **100**(469) 36–50.

Chan, Carri W, Galit Yom-Tov, Gabriel Escobar. 2014. When to use speedup: An examination of service systems with returns. *Operations Research* **62**(2) 462–482.

Delasay, Mohammad, Armann Ingolfsson, Bora Kolfal. 2015. Modeling load and overwork effects in queueing systems with adaptive service rates. *Working Paper* .

Delasay, Mohammad, Armann Ingolfsson, Bora Kolfal, Kenneth Schultz. 2016. Load effect on service times. *Working Paper, available at SSRN: http://ssrn.com/abstract=2647201 or http://dx.doi.org/10.2139/ssrn.2647201* .

Deo, Sarang, Aditya Jain, Pradeep Kumar Pendem. 2014. Pacing work in the presence of goals and deadlines: Econometric analysis of an outpatient department. *working paper* .

Do, Hung, Masha Shunko, Marilyn Lucas, David Novak. 2015. Can server behavior and queueing system design outweigh the benefits of pooling? *working paper* .

Dong, Jing, Pnina Feldman, Galit Yom-Tov. 2013. Slowdown services: Staffing service systems with load-dependent service rate. *Working paper* .

Doroudi, Sherwin, Ragavendran Gopalakrishnan, Adam Wierman. 2011. Dispatching to incentivize fast service in multi-server queues. *ACM SIGMETRICS Performance Evaluation Review* **39**(3) 43–45.

Edie, Leslie C. 1954. Traffic delays at toll booths. *Journal of the operations research society of America* **2**(2) 107–138.

Eppen, Gary D. 1979. Note-effects of centralization on expected costs in a multi-location newsboy problem. *Management Science* **25**(5) 498–501.

Feldman, P, J Li, GB Yom-Tov, E Yom-Tov. 2014. Service time sensitivity to load: Who is to blame? *Working paper* .

George, Jennifer M, J Michael Harrison. 2001. Dynamic control of a queue with adjustable service rate. *Operations Research* **49**(5) 720–731.

Gilbert, Stephen M, Z Kevin Weng. 1998. Incentive effects favor nonconsolidating queues in a service system: The principal–agent perspective. *Management Science* **44**(12-part-1) 1662–1669.

Jackson, James R. 1963. Jobshop-like queueing systems. *Management science* **10**(1) 131–142.

Jaeker, Berry, Jillian Alexandra, Anita Lynn Tucker. 2012. Hurry up and wait: Differential impacts of congestion, bottleneck pressure, and predictability on patient length of stay. *Harvard Business School Working Paper* .

Jain, Aditya, Sanjog Misra, Nils Rudi. 2014. Search, sales assistance and purchase decisions an analysis using retail video data. *Working paper* .

Karau, Steven J, Kipling D Williams. 1993. Social loafing: A meta-analytic review and theoretical integration. *Journal of personality and social psychology* **65**(4) 681.

Kc, Diwas S, Christian Terwiesch. 2009. Impact of workload on service time and patient safety: An econometric analysis of hospital operations. *Management Science* **55**(9) 1486–1498.

Kc, Diwas Singh, Christian Terwiesch. 2012. An econometric analysis of patient flows in the cardiac intensive care unit. *Manufacturing & Service Operations Management* **14**(1) 50–65.

Kleinrock, Leonard. 1976. Queueing systems, volume ii: Computer applications .

Krumm, Dianne. 2001. *Psychology at work: An introduction to industrial/organizational psychology*. Macmillan.

Lu, Yina, Aliza Heching, Marcelo Olivares. 2014. Productivity analysis in services using timing studies. *Available at SSRN 2403336* .

Lu, Yina, Andrés Musalem, Marcelo Olivares, Ariel Schilkrut. 2013. Measuring the effect of queues on customer purchases. *Management Science* **59**(8) 1743–1763.

Luskin, Mary Lee, Robert C Luskin. 1986. Why so fast, why so slow: Explaining case processing time. *J. Crim. L. & Criminology* **77** 190.

Mandelbaum, Avishai, Martin I Reiman. 1998. On pooling in queueing networks. *Management Science* **44**(7) 971–981.

Mas, Alexandre, Enrico Moretti. 2009. Peers at work. *The American Economic Review* **99**(1) 112–145.

Morgeson, Frederick P., Stephen E. Humphrey. 2008. *Job and team design: Toward a more integrative conceptualization of work design*. 39–91. doi:10.1016/S0742-7301(08)27002-7. URL `http://www.emeraldinsight.com/doi/abs/10.1016/S0742-7301%2808%2927002-7`.

Rothkopf, Michael H, Paul Rech. 1987. Perspectives on queues: Combining queues is not always beneficial. *Operations Research* **35**(6) 906–909.

Schultz, Kenneth L, David C Juran, John W Boudreau. 1999. The effects of low inventory on the development of productivity norms. *Management Science* **45**(12) 1664–1678.

Schultz, Kenneth L, David C Juran, John W Boudreau, John O McClain, L Joseph Thomas. 1998. Modeling and worker motivation in jit production systems. *Management Science* **44**(12) 1595–1607.

Shunko, Masha, Julie Niederhoff, Yarosla Rosokha. 2014. Humans are not machines: Impact of queueing design on service time. *Working paper* .

Song, Hummy, Anita L Tucker, Karen L Murrell. 2015. The diseconomies of queue pooling: An empirical investigation of emergency department length of stay. *Management Science* .

Stidham Jr., Shaler, Richard R. Weber. 1989. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations Research* **37**(4) 611–625.

Tan, Tom Fangyun, Serguei Netessine. 2014. When does the devil make work? an empirical study of the impact of workload on worker productivity. *Management Science* **60**(6) 1574–1593.

The World Bank. 2016. Services, etc., value added (% of GDP). URL `http://data.worldbank.org/indicator/NV.SRV.TETC.ZS`.

Whitt, Ward. 1999. Partitioning customers into service groups. *Management Science* **45**(11) 1579–1592.

Appendix A: Proofs

Proposition 3 establishes the result leading to Hypothesis 1(a).

PROPOSITION 3. *The average queue length is shorter in an M/M/2 queue with arrival rate* 2λ *and service rate* μ *than that in an M/M/1 queue with arrival rate* λ *and service rate* μ.

Proof of Proposition 3. Define $\rho = \lambda/\mu$, then the average queue length in an M/M/1 queue is $\frac{\rho^2}{1-\rho}$, and the average queue length in the M/M/2 queue is $\frac{2\rho^3}{1-\rho^2} = \frac{\rho^2}{1-\rho} \cdot \frac{2\rho}{1+\rho} < \frac{\rho^2}{1-\rho}$. The proof is complete. □

Proposition 4 establishes the result leading to Hypothesis 2.

Let the arrivals follow a stationary Poisson process with rate λ. As suggested by the service rate control literature (e.g. George and Harrison 2001), under some general conditions (e.g., increasing holding/waiting cost and increasing effort cost), the optimal exponential service rate is an increasing function of the queue length. Denote the queue-length (n) dependent service rates by $\mu_n, n = 0, 1, 2, ...$, then we have $\mu_0 \le \mu_1 \le \mu_2 \le \mu_3 ....$

Furthermore, let $t_n, n = 0, 1, 2, ...$ be a sequence of independent exponential random variables with rates $\lambda + \mu_n$, and define T_n to be the total random service time of a customer for which n is the length of the queue at the start of service. Then we have

$$T_n = \begin{cases} t_n & \text{w.p. } \frac{\mu_n}{\lambda+\mu_n} \\ t_n + t_{n+1} & \text{w.p. } \frac{\lambda}{\lambda+\mu_n}\frac{\mu_{n+1}}{\lambda+\mu_{n+1}} \\ ... & ... \end{cases}$$

PROPOSITION 4. $T_n \ge T_{n+1}, \forall n = 0, 1, 2...$ *in a stochastic dominance fashion, with the inequality being strict if* $\mu_m < \mu_{m+1}$ *for some* $m \ge n$.

Proof of Proposition 4. It is clear that $\mu_0 \le \mu_1 \le \mu_2 \le \mu_3 ...$ implies $t_1 \ge t_2 \ge t_3 \ge ...$ in a stochastic dominance fashion, for which any strict inequality in the first condition implies the corresponding inequality in the second condition.

From the definition of T_n we can write:

$$T_n = t_n + \frac{\lambda}{\lambda+\mu_n}t_{n+1} + \frac{\lambda}{\lambda+\mu_n}\frac{\lambda}{\lambda+\mu_{n+1}}t_{n+2} + \frac{\lambda}{\lambda+\mu_n}\frac{\lambda}{\lambda+\mu_{n+1}}\frac{\lambda}{\lambda+\mu_{n+2}}t_{n+3} + ...$$

$$T_{n+1} = t_{n+1} + \frac{\lambda}{\lambda+\mu_{n+1}}t_{n+2} + \frac{\lambda}{\lambda+\mu_{n+1}}\frac{\lambda}{\lambda+\mu_{n+2}}t_{n+3} + \frac{\lambda}{\lambda+\mu_{n+1}}\frac{\lambda}{\lambda+\mu_{n+2}}\frac{\lambda}{\lambda+\mu_{n+3}}t_{n+4} + ...$$

Comparing the two equations, we observe

1. The first term: $t_n \geq t_{n+1}$.
2. The second term: $t_{n+1} \geq t_{n+2}$ and $\frac{\lambda}{\lambda+\mu_n} \geq \frac{\lambda}{\lambda+\mu_{n+1}}$ because $\mu_n \leq \mu_{n+1}$.
3. And so on.

Therefore, $T_n \geq T_{n+1}$, with the inequality being strict if any of the inequalities in the $\mu_m, m \geq n$ relationship is strict. □

Proposition 5 is our main result that establishes the convexity of average service time.

PROPOSITION 5. *$E(T_n)$ is convex decreasing in n.*

Proof of Proposition 5. We establish some preliminary results first. Suppose $f_n \geq 0$ and $g_n \geq 0$ are both convex decreasing in n.

LEMMA 1. *$f_{n+1}g_{n+1} + f_n g_n \geq f_{n+1}g_n + f_n g_{n+1}$.*

Proof of Lemma 1. $f_{n+1}g_{n+1} + f_n g_n - (f_{n+1}g_n + f_n g_{n+1}) = (f_{n+1} - f_n)(g_{n+1} - g_n) \geq 0$. □

LEMMA 2. *$f_n g_n$ is convex decreasing in n.*

Proof of Lemma 2. It's easy to show $f_n g_n$ is decreasing in n. To show that it is convex, we observe the following:

$$
\begin{aligned}
f_{n+2}g_{n+2} + f_{n+2}g_n &\geq 2f_{n+2}g_{n+1} && \text{(convexity)} \\
f_{n+2}g_{n+1} + f_n g_{n+1} &\geq 2f_{n+1}g_{n+1} && \text{(convexity)} \\
f_{n+2}g_{n+1} + f_{n+1}g_n &\geq f_{n+2}g_n + f_{n+1}g_{n+1} && \text{(Lemma 1)} \\
f_{n+1}g_{n+1} + f_n g_n &\geq f_{n+1}g_n + f_n g_{n+1} && \text{(Lemma 1)}
\end{aligned}
$$

Adding all four together we get $f_{n+2}g_{n+2} + f_n g_n \geq f_{n+1}g_{n+1} + f_{n+1}g_{n+1}$. Therefore, $f_n g_n$ is convex in n. □

LEMMA 3. *$\frac{1}{\lambda+\mu_n}$ is convex decreasing in n.*

Proof of Lemma 3. It's easy to see $\frac{1}{\lambda+\mu_n}$ decreases in n because μ_n increases in n.

$$
\begin{aligned}
&\frac{1}{\lambda+\mu_n} + \frac{1}{\lambda+\mu_{n+2}} - \frac{2}{\lambda+\mu_{n+1}} \\
&= \frac{[(\lambda+\mu_{n+1})(\lambda+\mu_{n+2}) + (\lambda+\mu_n)(\lambda+\mu_{n+1}) - 2(\lambda+\mu_n)(\lambda+\mu_{n+2})]}{(\lambda+\mu_n)(\lambda+\mu_{n+1})(\lambda+\mu_{n+2})} \\
&= \frac{[\lambda(2\mu_{n+1} - \mu_{n+2} - \mu_n) + \mu_{n+2}(\mu_{n+1} - \mu_n) + \mu_n(\mu_{n+1} - \mu_{n+2})]}{(\lambda+\mu_n)(\lambda+\mu_{n+1})(\lambda+\mu_{n+2})}
\end{aligned}
$$

Since μ_n is concave increasing, all the terms in the numerator are positive, so $\frac{1}{\lambda+\mu_n}$ is convex in n. □

$$E(T_n) = E(t_n) + \frac{\lambda}{\lambda+\mu_n}E(t_{n+1}) + \frac{\lambda}{\lambda+\mu_n}\frac{\lambda}{\lambda+\mu_{n+1}}E(t_{n+2}) + \frac{\lambda}{\lambda+\mu_n}\frac{\lambda}{\lambda+\mu_{n+1}}\frac{\lambda}{\lambda+\mu_{n+2}}E(t_{n+3}) + ... \tag{8}$$

Since $E(t_n) = \frac{1}{\lambda+\mu_n}$, by Lemma 3 it is convex decreasing in n. Applying Lemma 3 one more time, we see all the individual terms in the coefficients of the $E(t_n)$ terms in (8) are also convex decreasing in n. It follows immediately from Lemma 2 that $E(T_n)$ is convex decreasing in n. □

Proof of Proposition 1. The average time in system in an M/M/1 with arrival rate λ and service rate μ ($\mu > \lambda$) is

$$W_{s1} = \frac{1}{\mu-\lambda};$$

In an M/M/2 with arrival rate 2λ and service rate μd ($d > \lambda/\mu$), the average time in system is

$$W_{s2} = \frac{\mu d}{(\mu d+\lambda)(\mu d-\lambda)}.$$

Define $\rho = \lambda/\mu$, then $W_{s1} < W_{s2}$ if and only if

$$f(d) \stackrel{\text{def}}{=} d^2 - (1-\rho)d - \rho^2 < 0.$$

Because $f(0) = -\rho^2 < 0$, $f(1) = \rho(1-\rho) > 0$, we deduce that

$$\bar{d}(\rho) = \frac{1-\rho+\sqrt{(1-\rho)^2+4\rho^2}}{2}$$

is the bigger quadratic root of $f(d) = 0$. Moreover, $0 < \bar{d}(\rho) < 1$ and $f(d) > 0$ if and only if $d > \bar{d}(\rho)$. Therefore, $W_{s1} < W_{s2}$ if and only if $d < \bar{d}(\rho)$. □

Proof of corollary 1.

$$\bar{d}'(\rho) = \frac{(5\rho-1) - \sqrt{5\rho^2-2\rho+1}}{2\sqrt{5\rho^2-2\rho+1}}.$$

So $\bar{d}'(\rho) > 0$ if and only if $(5\rho-1) > \sqrt{5\rho^2-2\rho+1}$, which is equivalent to $\rho > 2/5$. □

Proof of Proposition 2. In M/M/1, the balance equations are $P_i = P_{i-1}\frac{\lambda}{\mu_{i-1}}$, $i \geq 1$. Then, because $\sum_{i=0}^{\infty} P_i = 1$, we have

$$P_0 = \frac{1}{1+\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\cdots\mu_{i-1}}}.$$

Therefore, the average time in system in an M/M/1 is

$$W_{s1} = \frac{\sum_{i=1}^{\infty} iP_i}{\lambda} = \frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\cdots\mu_{i-1}}}{1+\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\cdots\mu_{i-1}}}.$$

Similarly, denoting $\mu_i' = d\mu_i$, we can derive the average time in system for an M/M/2 as

$$W_{s2} = \frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0'\mu_0'\cdots\mu_{i-2}'}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0'\mu_0'\cdots\mu_{i-2}'}}.$$

We then prove the following lemma.

LEMMA 4. $\frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}}$ *is decreasing in* μ_k *for all* $k\geq 0$.

Proof of Lemma 4. Given $k\geq 1$, define $a=\sum_{i=1}^{k+1}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}$, $b=\sum_{i=k+2}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}$, $c=1+2\sum_{i=1}^{k+1}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}$, $d=2\sum_{i=k+2}^{\infty}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}$. Then we have

$$f(\mu_k)=\frac{a+b/\mu_k}{c+d/\mu_k}=\frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}}.$$

Then

$$f'(\mu_k)=\frac{ad-cb}{(c\mu_k+d)^2},$$

which has the same sign as $ad-cb$.

$$\begin{aligned}
ad-cb=&\sum_{i=1}^{k+1}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}*2\sum_{j=k+2}^{\infty}\frac{\lambda^j}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{j-2}}\\
&-\sum_{i=k+2}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}*\left(1+2\sum_{j=1}^{k+1}\frac{\lambda^j}{\mu_0\mu_0\cdots\mu_{j-2}}\right)\\
=&\,2\sum_{j=k+2}^{\infty}\sum_{i=1}^{k+1}\left(\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}\frac{\lambda^j}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{j-2}}-\frac{j\lambda^{j-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{j-2}}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}\right)\\
&-\sum_{i=k+2}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}\\
=&\,2\sum_{j=k+2}^{\infty}\sum_{i=1}^{k+1}\frac{\lambda^{i+j-1}(i-j)}{\mu_0\mu_0\cdots\mu_{i-2}\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{j-2}}-\sum_{i=k+2}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}.
\end{aligned}$$

Because $j\geq k+2>i$, we have $2\sum_{j=k+2}^{\infty}\sum_{i=1}^{k+1}\frac{\lambda^{i+j-1}(i-j)}{\mu_0\mu_0\cdots\mu_{i-2}\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{j-2}}<0$, then

$$ad-cb<-\sum_{i=k+2}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{k-1}\mu_{k+1}\cdots\mu_{i-2}}<0.$$

Therefore, $f'(\mu_k)<0$, which means the $\frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}}$ is decreasing in μ_k for all $k\geq 1$. Similarly, we can also prove that $\frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu_0\mu_0\cdots\mu_{i-2}}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu_0\mu_0\cdots\mu_{i-2}}}$ is decreasing in μ_0. □

As d increases, all μ'_k increases, then $\frac{\sum_{i=1}^{\infty}\frac{i\lambda^{i-1}}{\mu'_0\mu'_0\cdots\mu'_{i-2}}}{1+2\sum_{i=1}^{\infty}\frac{\lambda^i}{\mu'_0\mu'_0\cdots\mu'_{i-2}}}$ decreases. So W_{s2} is decreasing in d.

When $d\to 0$, clearly, the time in system goes to infinity, that is, $\lim_{d\to 0}W_{s2}=\infty$. Given that W_{s2} is decreasing in d, there exists a unique $\hat{d}$, such that $W_{s1}<W_{s2}$ if and only if $d>\hat{d}$. □

Appendix B: Estimation Results of Queue Length Effect in Shared Queues

Table 8 presents the estimation results of Models (2) and (3) using transactions in shared queues. We find that the coefficient of queue length on service time is -0.0576 in Model (2), which implies that as the queue length increases by one, the service time decreases by approximately 5.76% on average. Also, this coefficient is statistically significant at the 1% significance level. This result provides support to Hypothesis 2. In Model (3), we allow a quadratic relationship between the queue length and the logged service time. The coefficients of $QueueLength_i^2$ and $QueueLength_i$ are 0.00888 and -0.109. The estimation results imply that the queue length induced speedup is diminishing as the queue length increases, which supports Hypothesis 3.

Table 8 Impact of Queue Length: Using Transactions in Pooled Queues

	(1)	(2)
VARIABLES	LogServiceTime	LogServiceTime
QueueLength	-0.0576**	-0.109**
	(0.00938)	(0.0198)
QueueLength^2		0.00888**
		(0.00282)
Log(NumberItems)	0.0972**	0.0966**
	(0.0231)	(0.0229)
Log(NumberGrocery)	0.106**	0.107**
	(0.0277)	(0.0275)
Log(TotalValue)	0.115**	0.116**
	(0.0119)	(0.0122)
Log(TotalSKU)	0.107**	0.106**
	(0.0296)	(0.0290)
Control Variables	Included	Included
Observations	1,060	1,060
R^2	0.464	0.469

Robust standard errors in parentheses
** p<0.01, * p<0.05

Appendix C: Estimation Results when Using Levels of Transaction Characteristics as Control Variables

We now examine several variations of the empirical models to test the robustness of our empirical results. First, we use levels of transaction characteristics variables $NumberGrocery_i$, $NumberItems_i$, $TotalValue_i$, and $TotalSKU_i$ instead of their natural logs, as control variables.

The corresponding regression models after such replacements for Models (2) and (3) are denoted as Models (2.1) and (3.1), whose estimation results are shown in the first two result columns of Table 9. From Table 9, we can see that the coefficients of $QueueLength_i$ and $QueueLength_i^2$ are negative and positive respectively, and their magnitudes are also similar to those reported in Table 4, which confirms that our results are robust. For example, the coefficient of $QueueLength_i$ in Model (2.1) is estimated to be -0.0485, which is similar to that in Model (2), -0.048. In addition, the R^2 are smaller in Models (2.1) and (3.1) than those in Models (2) and (3), which suggests that regressions using natural logs of transaction characteristics have better fit than those using levels of transaction characteristics. For example, the R^2 in Model (2.1), 0.44, is smaller than that in Model (2), 0.476. Models (2.1) and (2) have the same number of independent variables, and their only difference is how to use the transaction characteristics as control variables. So a higher R^2 implies a better fit.

Similarly, for the direct effect, we use level values of transaction characteristics to replace their natural logs in Model (4), and the resulting model is denoted as Model (4.1). The results are reported in the first column in Table 10. The coefficient of $SingleServerQueue_i$ is negative and significant, and the magnitude is also similar to that in Table 5. For example, the estimated coefficient of $SingleServerQueue_i$ in Model (4.1), -0.0968, is similar to that in Model (4), -0.107. Also, the interaction terms remain insignificant. In addition, the R^2 in Model (4.1), 0.435, is smaller than that in Model (4), 0.474. The comparison of R^2 suggest that regressions using natural logs of transaction characteristics have better fit.

The corresponding empirical model of Model (5) that uses level values of transaction characteristics to replace their natural logs is denoted as Model (5.1). The estimation result, as presented in the second column of Table 10, also confirms that the aggregate queue configuration effect on the service rate is slowdown.

We next use the service time itself, instead of its log value, as the dependent variable. We also use levels of transaction characteristics as control variables, as their impact on service time should be linear.

Table 9 Impact of Queue Length: Using Levels of Transaction Characteristics as Controls

VARIABLES	(2.1) LogServiceTime	(3.1) LogServiceTime	(2.2) ServiceTime	(3.2) ServiceTime
QueueLength	-0.0485**	-0.0973**	-3.719**	-7.896**
	(0.00709)	(0.0145)	(0.551)	(1.214)
QueueLength^2		0.0102**		0.876**
		(0.00240)		(0.180)
Transaction characteristics	Level value	Level value	Level value	Level value
Observations	3,245	3,245	3,245	3,245
R^2	0.440	0.444	0.384	0.388

Robust standard errors in parentheses
** p<0.01, * p<0.05

Table 10 Impact of Queue Configuration: Using Levels of Transaction Characteristics as Controls

VARIABLES	(4.1) LogServiceTime	(5.1) LogServiceTime	(4.2) ServiceTime	(5.2) ServiceTime
SingleServerQueue	-0.106**	-0.0667*	-10.20**	-6.076*
	(0.0327)	(0.0270)	(3.268)	(2.538)
QueueLength	-0.115**		-10.19**	
	(0.0180)		(1.530)	
QueueLength^2	0.00972**		0.913**	
	(0.00287)		(0.213)	
SingleServerQueue *QueueLength	0.0234		2.574	
	(0.0215)		(1.879)	
SingleServerQueue *QueueLength^2	-0.000508		-0.0627	
	(0.00345)		(0.270)	
Transaction characteristics	Level value	Level value	Level value	Level value
Observations	4,305	4,305	4,305	4,305
R^2	0.477	0.456	0.386	0.366

Robust standard errors in parentheses
** p<0.01, * p<0.05

The corresponding regression models of Models (2.1) and (3.1) after replacing the dependent variable are denoted as Models (2.2) and (3.2). The estimation results are presented in the last two columns in Table 9. The coefficient for $QueueLength_i$ is negative and significant, which means that the service time decreases as the queue length increases, thereby supporting our Hypothesis 2. The interpretations of coefficients are different from those in Section 5.1 because of the difference in the dependent variable used. Further, the magnitudes of the estimated coefficients are not directly comparable with those in Section 5.1. For example, the coefficient of $QueueLength_i$ in Model (2.2), -3.719, means that as the queue length increases by one, the service time is shortened by 3.719 seconds on average. However, after transforming the impact into percentage change in service time, it is comparable to results in Section 5.1. The average service time is 77.32 seconds, so a decrease of 3.719 seconds corresponds to a $3.719/77.32 \approx 4.81\%$ decrease in service time, which is similar to the estimated 4.8% decrease in Model (2). In addition, the estimated coefficient for $QueueLength_i^2$ is positive and significant, which implies that the marginal effect of queue length on service time is decreasing and thus supports Hypothesis 3.

For the direct effect, replacing the dependent variable in (4.1) with service time itself, we have model (4.2), whose estimation result is shown in the third column of Table 10. The estimated coefficient for $SingleServerQueue_i$ remains negative and significant, and supports Hypothesis 4(a), which is consistent with the result in Section 5.2. In addition, even though the values of the estimated coefficient for $SingleServerQueue_i$ in Table 10 are not directly comparable with those in Table 5 because of the different interpretation, the implied percentage changes in service time are similar. For example, the estimated coefficient of $SingleServerQueue_i$ in Model (4.2) is -9.935, which means that transactions in dedicated queues are about 9.935 seconds faster than those in shared queues. Given the average service time being 77.32 seconds, a 9.935 seconds difference corresponds to $9.935/77.32 \approx 12.85\%$, which is similar to 10.7%, the result in Model (4). In addition, the interaction terms remain insignificant.

Replacing the dependent variable in (5.1) with service time itself, we have model (5.2). The estimation result, as presented in the fourth column of Table 10, also confirms that the aggregate queue configuration effect on the service rate is slowdown.