Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

DELAY GUARANTEE PLANNING OF CALL-BACK OPTIONS IN TIME-VARYING SERVICE SYSTEMS

Galit B. Yom-Tov
Tali Zeitler

Faculty of Industrial Engineering and Management
Technion—Israel Institute of Technology
Technion City
Haifa, 3200003, ISRAEL

ABSTRACT

Many service centers offer a "call-back" option, in which customers entering the queue are informed of the anticipated (online) wait and can choose to wait either online or off-line till an agent contacts them. We show that such a policy has the potential of both improving service performance and server utilization, by balancing the load between overloaded and underloaded periods. However, our analysis suggests that companies need not offer that service at all times, and that the delay guarantees proposed should be planned according to the anticipated load throughout the day. In order to optimize the operation of such a system, we develop an Iterative Simulation Algorithm to determine what delay guarantees the company should offer in a time-varying environment. Those guarantees depend on service level targets the company wishes to provide and the delay sensitivity of the online vs. off-line customers.

1 INTRODUCTION

Many call centers nowadays offer their customers a call-back option. The customers, who arrive to the queue, are informed about the anticipated waiting time and the maximal delay till call-back. They can then choose between waiting online in the regular queue or waiting offline for a call that will be initiated by the organization. The offline wait is generally longer than a real time wait, which we would refer to as an "online" wait, but enables the customer to do other things while waiting, and hence, some customers prefer it. The customer decides whether to choose this option according to information provided regarding the waiting time in queue as well as the maximal delay guarantees for the amount of time he would wait for the call-back. By giving such a call-back option, the firm can balance loads and reduce waiting time for online customers.

In many cases, the maximal offline delay guarantee is constant, offered to all customers and independent of the system's load. For example, big companies in Israel are required by law to offer a call-back option if the online wait is longer than 3 minutes and, if such an option is chosen, to call-back the customers within 3 business hours. Such a policy is in the interest of the company, as it has the potential of both improving service performance, and server utilization, by balancing load between overloaded and underloaded periods. The problem with the current practice is twofold: 1. It ignores the fact that the length of offline delay influences customer choice, i.e. customers are sensitive to the online and offline delays. 2. It also ignores the daily pattern of load, and may move customers from overloaded periods to even more overloaded ones, causing worse delays instead of improving them. Therefore, planning delay guarantees that vary over time is important. In this work, we provide a simulation-based optimization algorithm, called the Iterative Delay Guarantee (IDG) algorithm, to determine what offline waiting guarantees a company should offer in a time-varying environment with customer choice. Those guarantees change over time and depend on the load of the current state of the system as well as the anticipated load of future time periods.

Though the call-back option exists in most service centers nowadays, the amount of research considering the operations of service systems with call-backs is quite limited. Armony and Maglaras (2004b) studied call-back in a stationary system where all customers are offered the same delay guarantee. They showed that applying call-back improves system performance in stationary systems and reduces customers' online waiting time. The call-back option basically smooths the stochastic fluctuation of the workload. Armony and Maglaras (2004a) studied the value of providing real-time delay information in the setting of Armony and Maglaras (2004b). Legros et al. (2016) studied when to offer call-back using a Markov decision process and characterized a threshold policy. Ata and Peng (2017) studied when to offer call-back under arrival rate uncertainty and the ability to look a bit into the future.

Our research focuses on a time-varying system in which arrival rates change throughout the day and the system alternates between overloaded and underloaded periods. Such alternation is caused by a mismatch between service demand and capacity usually due to some constraints, and manifests itself in long queues and customer waiting. The main focus of this work is to determine the appropriate delay guarantees for systems to achieve specific service level goals for the online waiting times. We take into account the influence on the customer's choice. The longer the due date, the less attractive the alternative call-back option for the customers becomes. We suggest that in order to utilize the benefits of the call-back option properly, one should change the delay guarantee according to the anticipated state of the service system during all the hours of a normal working day. This is in contrast to current practice of a long constant delay guarantee regardless of current and future loads.

We use a simulation-based approach to determine the appropriate delay guarantee in order to reduce the average waiting time in the online queue to a certain predetermined threshold. We call our approach IDG - Iterative Delay Guarantee algorithm. It is inspired by the ISA (Iterative Staffing Algorithm) of Feldman et al. (2008). ISA basically uses simulation to check how staffing decision variables affect performance and adjusts them accordingly till reaching the targeted service level or the maximal iteration number, or to a point where the change in performance after each iteration is very minor. The use of simulation allows us to include realistic features that are fitted from real call center data in our model.

2 MODEL DESCRIPTION AND CALIBRATION

We model the service system as a variation of the time-varying Erlang-A queueing system. Customers arrive to the system according to a non-stationary Poisson process with rate $\Lambda(t)$. At the time of their arrival, t, they receive information regarding the anticipated online wait for service, $W(t)$ along with a maximal delay guarantee, $D(t)$. The customers then make their decision as whether to join the online queue or the call-back queue. Customers who prefer to wait online have finite patience which is exponentially distributed with rate θ. If the online wait exceeds the customer's patience, he/she abandons the online queue. There are $N(t)$ statistically identical servers at time t. Service times are assumed to be exponentially distributed with rate μ. Figure 1 provides a pictorial illustration of the model.

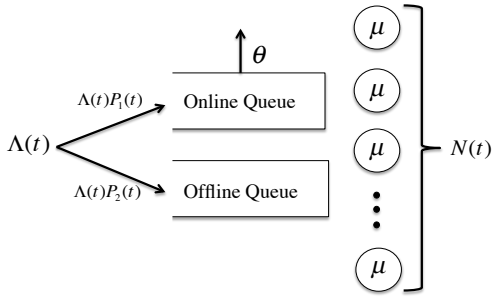


Figure 1: The queueing model.

Customer Choice Following Armony and Maglaras (2004b), we model the customer's decision according to a multinomial logit (MNL) choice model. We denote $u_1(t) = r - c_1 W(t) + \xi_1$ as the utility of the online option and $u_2(t) = r - c_2 D(t) + \xi_2$ as the utility of the offline (call-back) option. Particularly, r is value of service. c_1 is the online waiting cost and c_2 is the offline waiting cost. We assume $c_1 > c_2$ as the offline wait is much more pleasant compared to the online wait. ξ_i's are the random terms capturing the unobserved factors. We assume ξ_i's are i.i.d. Gumbel. This implies that for a customer arriving at time t, the probability of choosing to wait online is

$$P_1(t) = \frac{e^{r-c_1 W(t)}}{e^{r-c_1 W(t)} + e^{r-c_2 D(t)}} = \frac{e^{-c_1 W(t)}}{e^{-c_1 W(t)} + e^{-c_2 D(t)}},$$

and probability of asking for call-back is $P_2(t) = 1 - P_1(t)$.

Customer preferences to the different waiting options are important factors as they affect dramatically the effectiveness of the call-back options. Customers with low sensitivity to waiting costs tend to stay and wait online regardless of the announced waiting times. That fits a situation in which customers wait in queue for an important service that cannot be postponed, such as medical consultation or a call to the stock exchange teller. On the other hand, when customers are highly sensitive to online waits, they are more willing to postpone their service. In such situations, by applying call-backs, the system can become more efficient in reducing the online waits as the customers' decisions highly depend on suggested delay guarantees. For example, using the case-study data that will be described later and a constant delay of 3 hours, if $c_1 = c_2$ the expected online waiting time of served customers is 66 minutes, while if $c_1 = 5c_2$ ($c_1 = 15c_2$), the online waiting time reduces to 9.7 (3) minutes, respectively. Other performance measures, such as probability of abandonment and the probability of immediate service, also improve as that ratio grows. Companies may influence the ratio between online and offline waiting costs by changing the environment of the online wait.

Delay Estimator We use the head-of-line (HOL) delay as our delay estimator. The HOL delay estimation method basically announces the elapsed waiting time of the first customer in queue, who is next to enter service, to the latest arriving customer. HOL delay is simple to implement and is shown to be accurate in various applications (Ibrahim and Whitt 2011, Senderovich et al. 2014). It provides more up-to-date information among historical average-based estimators, and thus avoids delayed feedback in systems with customer choice (Dong et al. 2018).

Routing Rules The routing rule proposed by Armony and Maglaras (2004a) assumed a constant delay garantee. Therefore, it cannot be implemented directly. As Armony and Maglaras (2004a) explained, in an appropriate routing rule for call-back systems, the servers should prioritize offline queue customers if the delay constraint may be violated. Hence, we suggest the following routing rule: serve the online queue, unless there is a customer whose delay guarantee is about to be violated, i.e. at time t, the customer's delay guarantee is less than or equal to $t + 1/(N(t)\mu)$. This rule is based on the assumption that service times are i.i.d. with total rate $N(t)\mu$ at time t. In such a case, the next agent who becomes available serves the call-back queue customer. If the online queue is empty and there are customers in the call-back queue, they will be served before their due time. Specifically, let A_i be the arrival time of customer i. If this customer chooses the call-back option, he/she should be served before $A_i + D(A_i)$. Customers in the call-back queue are sorted by increasing delay guarantees; thus we switch service priorities in favor of call-back customers whenever the first customer's due time is less than or equal to $t + \frac{1}{N(t)\mu}$.

2.1 Model Calibration: Case Study from an Israeli Call Center

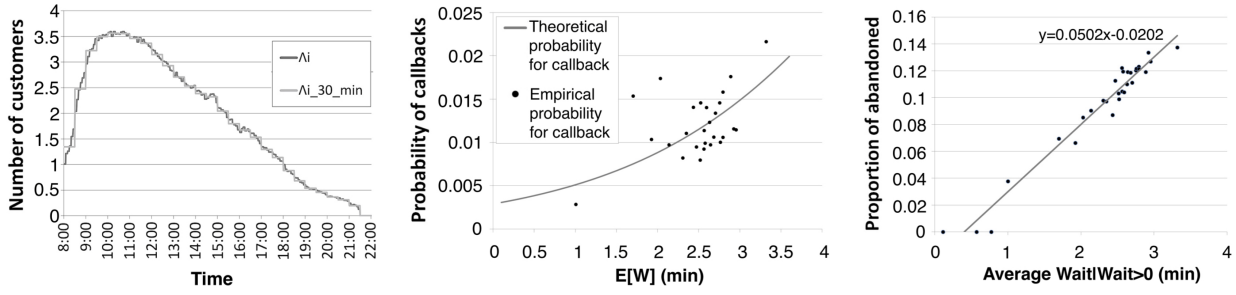
In this work we use a call center case study to demonstrate our approach. It is based on data from a medium size Israeli bank call center which offers a call-back option. The data was generously made available to us by the SEELab at the Technion.

2.1.1 Basic Data

Each call has several components: the interactive voice response (IVR) component, the queue component where the customer waits to speak with an agent, and the service component where the customer is served by the agent. The data captures the starting time, ending time, type of call— incoming/call-back—and outcome (e.g. abandoned, choice of call-back, etc.). The sequence of events for a customer is usually as follows: First the customer reaches the IVR. If the problem is answered by the IVR, the call is then terminated. Otherwise, the customer waits in queue until an agent becomes available. While the customer waits in queue, he/she might be provided with information regarding the estimated waiting time in the online queue. The data does not have information about the exact content of the system announcements. To the best of our knowledge, at the moment the customer enters the queue he/she is offered the call-back option with a constant delay guarantee of 3 hours. If the customer chooses to stay and wait in the online queue, that call-back option is offered again periodically while waiting. In our model, we make the simplified assumption that the call-back decision is made at the entrance to the system. From the data we observe 4% of the customers choose the call-back option. The average time till customers are called back is 54 min. On the other hand, 6% of the call-backs are made after more than 3 hours.

2.1.2 Model Fitting

The call center is open from 8:00 to 22:00 during weekdays. We divide the daily time horizon into 28 half-hour time intervals. We assume that the arrival rate is piecewise constant. For each interval i, we fit the arrival rate Λ_i using the average arrival count. We also assume that the number of servers, N_i, is fixed in each interval. By utilizing the differences in the HOL delay, we estimate (c_1, c_2) by fitting a logistic regression model. The fitted values are $c_1 = 0.48, c_2 = 0.031$, which suggest that the customers are about 15 times more sensitive to online wait than offline wait. To estimate the abandonment rate θ, we fit a simple linear regression model for the probability of abandonment using the average waiting time, taking into account only the customers who waited (Mandelbaum and Zeltyn 2004). The slope of the line, which is our estimate of θ, is $1/20$, i.e. the average patience time is 20 min. Figure 2 provides some details of our fitted model. The mean service time is 4.16 min.



(a) Fitting the piecewise constant arrival rate function.

(b) Fitting the probability of choosing the call-back option.

(c) Fitting the abandonment rate.

Figure 2: Case study data and parameter estimation for the model.

2.1.3 Settings

To test how the call-back option can help to balance the load, we set the number of agents so that the system will alternate between overloaded and underloaded periods. (If there is a perfect match between offered load and capacity, the queueing system will behave as if in steady state (Yom-Tov and Mandelbaum 2014); in such situations a constant delay guarantee such as the one suggested by Armony and Maglaras (2004a) is expected to perform very well.)

From now on, we will assume that each time interval i has a different delay guarantee D_i, which is constant throughout the interval. The service level is specified so that the maximal waiting time in the online queue shall stay below a desired threshold T_w. Hence, our performance target is $\max_i\{W_i\} \leq T_w$, where W_i is the *average* waiting time of the online queue on time interval i. Our goal is to find appropriate delay guarantees of the call-back option, D_i's, that achieve the desired service level.

3 THE ITERATIVE DELAY GUARANTEE ALGORITHM

In this section, we introduce an iterative algorithm to find the appropriate delay guarantees. We denote $D(i,j)$ as the delay guarantee of the call-back option in period i after the j-th iteration. Set $\alpha \in (0,1)$ as the decreasing rate. We also set

$$d_{ij} = \begin{cases} \max\left\{W_i, \frac{1}{c_2}\log\left(\frac{N_i\mu}{\Lambda_i - N_i\mu}\right) - \frac{c_1}{c_2}W_i\right\} & \text{if } \Lambda_i > N_{i+d_{ij}}\mu, \\ W_i & \text{if } \Lambda_i \leq N_{i+d_{ij}}\mu, \end{cases} \tag{1}$$

as the lower bound for D_i. This lower bound is set for two purposes. 1. We want to make sure the offline delay is longer than the online delay, as the company would not offer call-back that is shorter than the online waiting. Hence, $d_{ij} \geq W_i$. 2. We want to make sure we do not commit to a delay guarantee we cannot hold; that could happen if we overload the offline queue, i.e. $N_{i+d_{ij}}\mu < \Lambda_i e^{-c_2 d_{ij}}/(e^{-c_2 d_{ij}} + e^{-c_1 W_i})$. From this constraint we extract d_{ij}, as an additional bound in (1).

We start by initializing $D(i,0)$ to be some very large number such that even for very large values of delay (e.g. ten times more than the average service time), customers will still choose to wait online with probability close to 1. In each iteration, we run 150 replications of the queueing model over a two-day time horizon; each day consists of a 14-hour time interval (8:00–22:00). That time horizon is divided into I equal intervals. (In the simulations we show here time intervals are of 30 minutes.) We then check for each time interval if performance targets are met. In iteration j, if the online delay target is violated in interval i, we set $D(i,j) = \max\{\alpha D(i,j-1), d_{ij}\}$; otherwise, we set $D(i,j) = \max\{D(i,j-1), d_{ij}\}$. In the latter case, we reduce the offline delay guarantee to encourage more people to wait offline. We terminate if performance targets are met for all time intervals or we run out of computational budget. we summarize the iterative algorithm below.

The Iterative Delay Guarantee Algorithm
Input: Maximum number of iterations: J; Online delay threshold: T_w; Decreasing rate: α.
Output: Offline delay guarantees: D_i, $i = 1,2,\ldots,I$.
Initialization: Set $j = 0$, $D(i,j) = 1000$, and run simulation to get W_i's for $i = 1,2,\ldots,I$.
Step 1. Set $j = j+1$.
Step 2. Calculate d_{ij}.
Step 3. For each interval i, $i = 1,...,I$: If $W_i \leq T_w$, set $D(i,j) = \max\{D(i,j-1), d_{ij}\}$; Otherwise, i.e. if $W_i > T_w$, set $D(i,j) = \max\{\alpha D(i,j-1), d_{ij}\}$.
Step 4. Run simulation with updated $D(i,j)$'s to get W_i's for $i = 1,2,\ldots,I$.
Step 5. If $\max_i W_i > T_w$ and $j < J$, go back to step 1; Otherwise, output $D(i,j)$ for $i = 1,2,\ldots,I$.

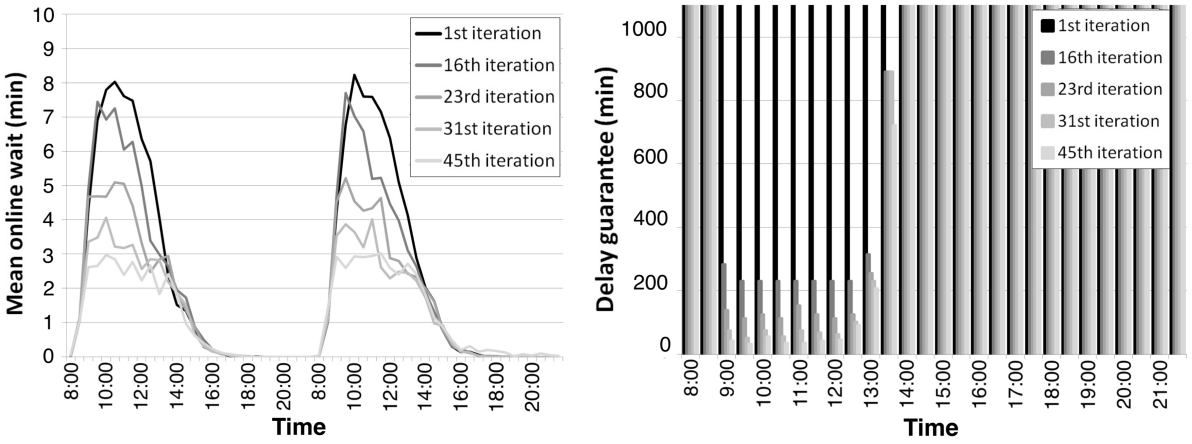
3.1 Simulation Results for the IDG Algorithm

We next examine the IDG algorithm for the simulation system calibrated in Section 2.1. We set the target average online delay at 3 minutes. In addition, to the average online delay in each time interval, we also look at other performance measures, including i) $E[W|Sr]$: the overall mean waiting time of the online queue customers who didn't abandon the queue; ii) $P(ab)$: the proportion of online queue customers abandoning the queue; iii) $P(cb)$: the proportion of customers choosing the call-back option; iv) $E[W_{cb}]$: the average

offline wait; v) RMSE:$=\sqrt{\frac{1}{I}\sum_{i=1}^{I}(W_i - E[w|Sr])^2}$, which measures the balance of performance during the day and how close they are to the average waiting time.

Figure 3 shows in detail that the mean waiting times converge towards the targeted value of 3 minutes (left diagram) and the corresponding changes in the values of delay guarantees (right diagram). The simulation started without call-backs (1st iteration) where the maximal average wait experienced by the online customers is about 8 minutes around 11:00 AM as seen in Figure 3(a); those values converge nicely to the target. The delay guarantee on Figure 3(b) demonstrates that call-backs are much more essential through overloaded periods while underloaded periods may not need call-backs at all.

Table 1 shows the daily average performance measures along IDG algorithm iterations. At the first iteration of the algorithm, all customers are informed with an initial and identical D_i of 1100 minutes. This value is high enough so that even if customers are expected to wait one hour in the online queue, the probability of choosing the call-back option is less than 0.5%. In such a case, the system acts as if there is no call-back option at all. Hence, the abandonment probability is 0.17 and the mean waiting time for the online customers is 2.38 minutes with RMSE of 25.34. At the final iteration we see improvement in all performance measures. The mean waiting times have decreased to 1.10 minutes and became more balanced as RMSE decreased to 9.50. The proportion of abandonment is 0.030 and the call-back rate is now 32.61%. By reducing delay guarantees, the value of $E[W_{cb}]$ decreases as well. In summary, we see that the algorithm managed to reduce waiting time, and they converge to the proposed performance target time during highly loaded periods.



(a) Average online delays.

(b) Offered offline delay guarantees.

Figure 3: Mean waiting times for online queue customers and delay guarantee values in minutes for different iterations of the IDG algorithm.

Table 1: Change in performance measures along IDG algorithm iterations.

Iteration number	$E[W\|Sr]$ (min)	$P(ab)$	$P(cb)$	$E[W_{cb}]$ (min)	RMSE
1	2.54	0.173	0	-	27.87
16	2.27	0.092	0.145	24.87	20.03
23	1.65	0.055	0.259	19.34	15.91
31	1.33	0.036	0.290	17.02	11.46
45	1.10	0.030	0.326	13.68	9.50

4 SENSITIVITY ANALYSIS

To conduct sensitivity analysis, we use a stylized sinusoidal arrival rate function: $\lambda(t) = R + \gamma \cdot sin(\omega t)$, $0 \leq t \leq T$, with $R = 50, \gamma = 10, \omega = 1$. For simplicity, we set the service rate μ and the patience rate θ both to 1, meaning the service time is about $\frac{1}{6}$ of a full cycle. As $\mu = 1$, R measures the average workload. We assume the ratio among online and offline waiting costs is 15 (customers are indifferent between waiting 1 minute on line or 15 minutes offline), similarly to our case study.

We demonstrate the effectiveness of the IDG algorithm by comparing the performance of the delay guarantees generated by the IDG algorithm to other methods. In particular, we conduct comparisons of the following delay guarantee policies: a) no call-backs; b) a constant delay guarantee of 180 minutes; c) a constant delay guarantee of 60 minutes; d) locally optimal solution in which we determine D_i for each interval ignoring the load in later periods, i.e., each time interval is considered as a stationary system of its own. This stationary delay guarantee was motivated by Armony and Maglaras (2004b); adjustments were done to incorporate abandonment instead of balking; e) the IDG delay guarantees.

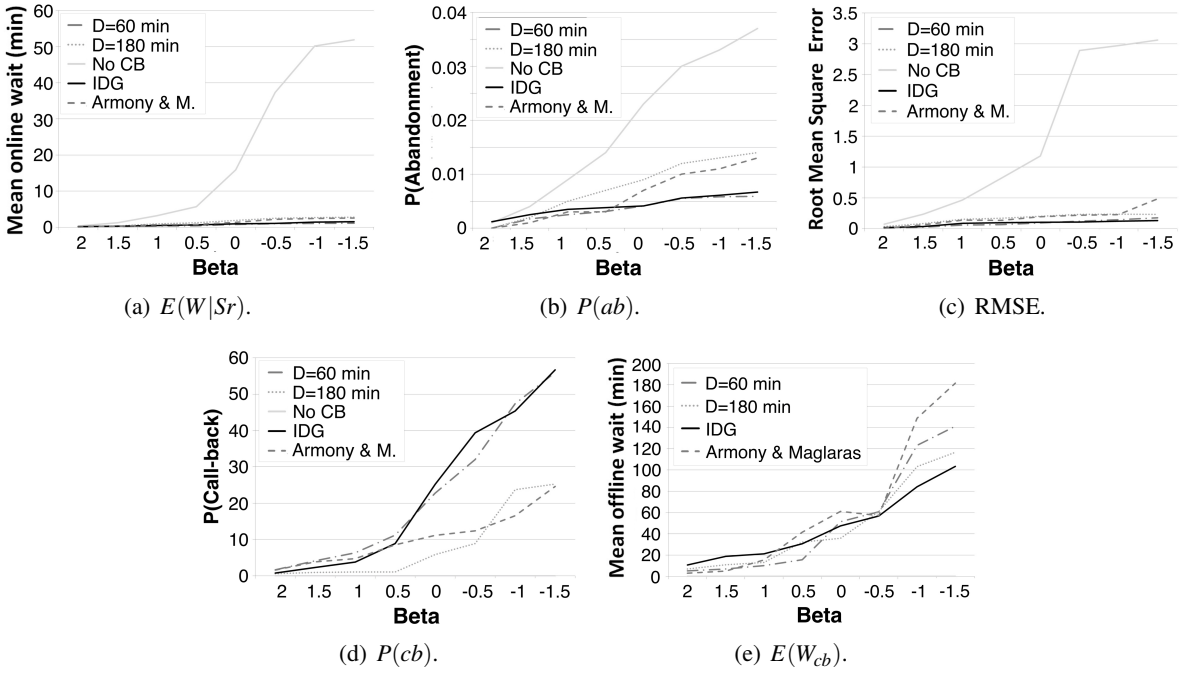
4.1 Change in Staffing Level

In this section, we explore how different call-back policies work for different staffing levels. The system we simulate alternates between overloaded and underloaded periods. We apply the square root staffing policy, i.e. the staffing level $n := R + \beta\sqrt{R}$, as an overall staffing decision, where $R = \Lambda E[S]$ is the average of total offered load over a day. We change the staffing level by varying the square root staffing parameter β. Large (positive) β will result in a mostly underloaded system that has a brief overloaded period, while small (negative) β creates a mostly overloaded system that has a brief underloaded period. Though call-back helps in both, we expect precise planning of call-back to be more important in the latter.

Figure 4 summarizes some of the performance measures as a function of β. In general, as β decreases, all performance measures deteriorate since resources are scarcer. This is clearly evident in the no CB option. However, having the call-back option and setting the right delay guarantee for each time interval improve system performance dramatically. When the delay guarantee is fixed, having a shorter delay guarantee improves performance, i.e. no call-back is outperformed by a constant delay guarantee of 180 min, which is outperformed by the constant D of 60 min. Setting a locally optimal D (Armony and Maglaras) performs surprisingly well. That method is only slightly worse than the IDG algorithm for most performance measures; differences are more apparent for lower values of β. We contribute this to the fact that the former does local optimization only, and doesn't take into account the effect of a certain due date suggestion on the load in the interval for which the customers are referred to. For this example, it seems from these graphs, that the constant delay guarantee of 60 minutes achieves a similar average performance as the IDG algorithm. However, as shown in Figure 5, though the average wait for $D = 60$ is less than for the IDG algorithm, the constant D can't assure any upper bound for online waiting. Note that, as opposed to the IDG algorithm, other delay guarantee setting methods do not take into account any targeted threshold as an input. Thus, if we want to promise our customers that whenever they call, their average online wait won't exceed a certain amount of time, we must use the IDG algorithm. Furthermore, the constant delay methods do not check if the service center is capable of returning the call-back within the promised delay guarantee, as shown in Figure 4(d); when we set D at a constant level of 60 min, call-backs are made after more than two hours for $\beta < -0.5$. This is problematic, if the system promises customers a specific delay guarantee it should hold to it, otherwise, it may harm customer trust in the system.

4.2 Change in Cycle Length

In this section, we explore the impact of changes in cycle length ω. When we increase ω, the system alternates between overloaded and underloaded periods more frequently. Note that IDG is the only method that takes a global viewpoint. In particular, the fixed D ignores cycle length, and Armony and Maglaras method takes the local optimization approach. We conjecture that the local optimization approach may not



(a) $E(W|Sr)$. (b) $P(ab)$. (c) RMSE.

(d) $P(cb)$. (e) $E(W_{cb})$.

Figure 4: Performance measures as a function of β.

perform well when the cycle length is small relative to D; in such cases, customers may be transferred from a moderately-loaded time period to a highly-loaded time period.

We compare performance measures, for the different delay guarantee policies, for $\omega = 0.5$, $\omega = 1$, and $\omega = 2$. We set the staffing level at 50. Simulation results in Table 2 show that in general, performance measures improve for all delay setting methods as ω increases, but the IDG method has an advantage over the other methods. For $\omega = 2$ the IDG algorithm outperforms all other delay setting methods for mean online wait, mean offline wait, and probability of immediate answer.

In order to understand better the differences between the policies, we examine in Figure 6 how the delay guarantees fluctuate over a full arrival rate cycle. We first observe that the IDG algorithm in general sets the delay guarantee in an opposite pattern to the one of the offered load, i.e. high delay guarantees are set during underloaded periods while low delay guarantees are set in overloaded periods. This does not happen at all in the Fixed D policy, and to a lesser extent in the Armony and Maglaras stationary policy. In more detail, when $\omega = 2$, if we look at the first time interval in Figure 6(a) for example, we see that a delay guarantee of 60 minutes is offered by both the Fixed 60 minutes and Armony and Maglaras policies. Consequently, customers who choose the call-back option should be served within the first hour although the offered load increases during that time. On the contrary, a 100-minute delay guarantee is offered by the IDG algorithm where customers should be served within a bit more than 1.5 hours and the offered load after 1.5 hours is lower than at the first time interval. Furthermore, when looking at Figure 7(a), showing the mean online waiting times for each time interval and each delay guarantee policy when $\omega = 2$, we see indeed that the mean online waiting time at the first time interval is a bit higher for the IDG algorithm but at the second time interval it clearly outperforms both the Fixed 60 and Armony and Maglaras. Another example is for the fifth time interval, where the IDG delay guarantee is quite long (over 3 hours) while Armony and Maglaras and Fixed 60 delay guarantees are much shorter (2 hours and 1 hour accordingly). Thus, more customers choose the call-back option for Armony and Maglaras and Fixed 60 methodologies although the system is in an underloaded period and furthermore, 1–2 hour delay guarantees transfer the customers to another time interval where the offered load is higher.

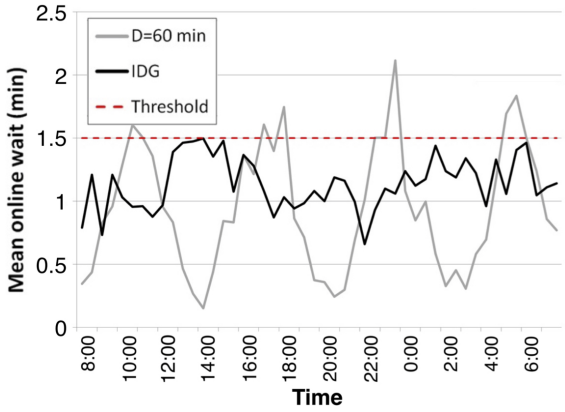
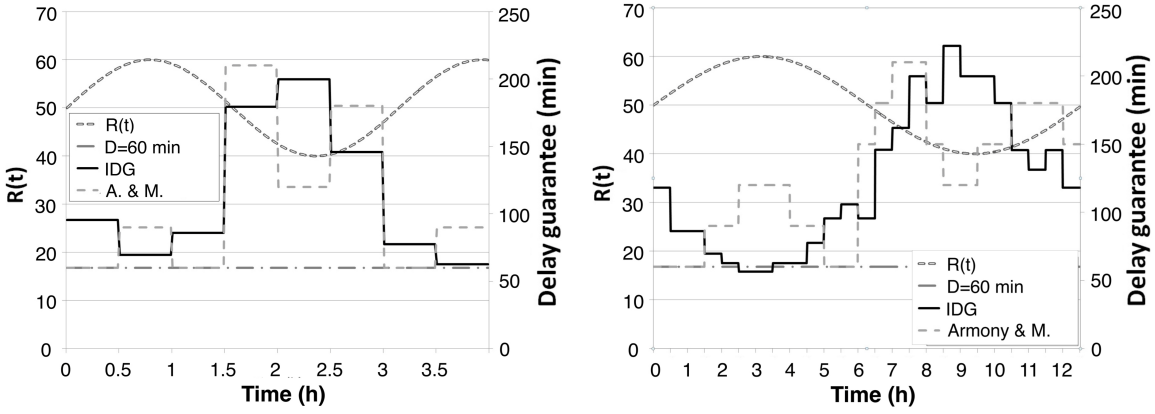


Figure 5: Mean online wait for $\beta = -0.5$ and 1.5-minute threshold, comparing IDG algorithm and $D = 60$ delay guarantee setting methods.

Figure 6(b) shows the different delay guarantee values for different time intervals along a cycle when $\omega = 0.5$. If we look at the first time interval again, we see that the Fixed 60 and Armony and Maglaras offer relatively short delay guarantees compared to the IDG algorithm. Consequently, customers would choose call-backs with higher probabilities for the former delay setting methods and will be delayed for 1 hour. During this hour the offered load increases. At the end of this hour, the delay guarantee of Armony and Maglaras increases. Thus, less customers choose the call-back option at the busier time interval as opposed to the IDG algorithm which delays less customers to a busier time interval but then offers a shorter delay guarantee in order to balance the loads. Figure 7 shows the mean online waiting times for each delay guarantee policy when $\omega = 0.5$, although the IDG is generally slightly outperformed by the Fixed 60 policy, it is clearly showing the difference between the two methods regarding the threshold. While the IDG algorithm assures all customers that the wait, on average, is less than 1.5 minutes, the Fixed 60 has no such feature. As for the Armony and Maglaras method, we see that many customers were transferred to overloaded time intervals which causes long waits between time intervals 4–12. The Fixed 60 policy performs quite well for all ω values as it simply offers a good alternative for the customer at any given moment. Thus, even if customers are transferred from an underloaded period to an overloaded period, the call-back option will be chosen again during the overloaded period and this may "balance" the damage.

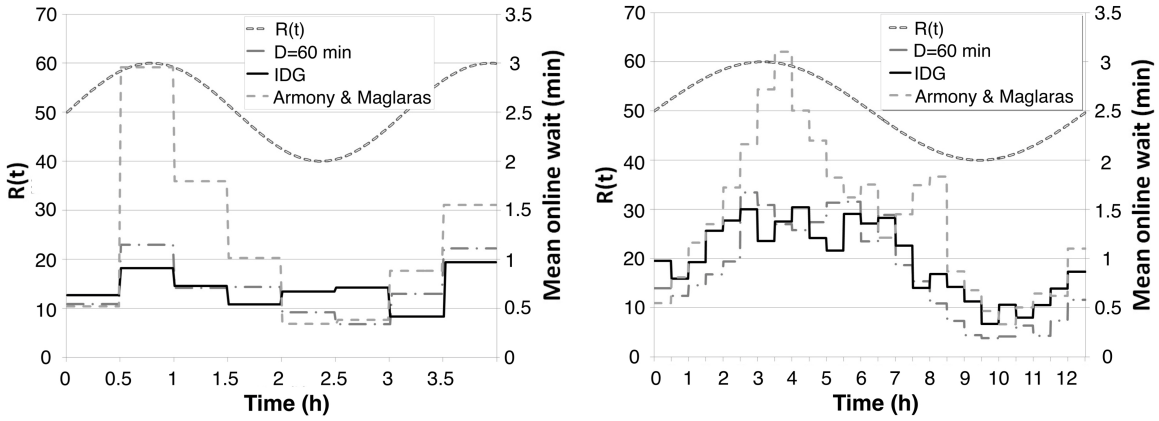


(a) $\omega = 2$. (b) $\omega = 0.5$.

Figure 6: Delay guarantee values in minutes for different delay guarantee setting methods and different values of ω.

Table 2: Performance measures when staffing under different values of ω, using different delay guarantee setting methods.

Delay guarantee method	ω	$E[w\|Sr]$ (min)	%CB	$P(ab)$	RMSE	$E[W_{cb}]$ (min)	$P(w=0)$
No call-backs	0.5	17.344	0	0.023	1.849	-	0.322
	1	15.902	0	0.022	1.169	-	0.351
	2	15.183	0	0.020	0.867	-	0.355
Fixed 180 min	0.5	1.983	6.43	0.011	0.224	43.86	0.541
	1	1.924	5.81	0.009	0.203	34.93	0.546
	2	1.919	4.63	0.010	0.174	29.14	0.559
Fixed 60 min	0.5	0.869	22.65	0.004	0.094	47.78	0.433
	1	0.761	21.84	0.004	0.087	50.93	0.450
	2	0.722	13.21	0.004	0.085	38.59	0.571
Armony and Maglaras	0.5	1.448	15.51	0.009	0.216	87.23	0.505
	1	1.395	12.02	0.008	0.188	64.90	0.536
	2	1.207	6.20	0.006	0.136	60.92	0.609
IDG algorithm	0.5	1.038	27.53	0.005	0.127	58.01	0.322
	1	0.954	24.97	0.004	0.103	44.33	0.466
	2	0.651	10.44	0.004	0.097	25.56	0.628



(a) $\omega = 2$. (b) $\omega = 0.5$.

Figure 7: Mean online waiting times in minutes for different delay guarantee setting methods and different values of ω.

5 CONCLUSION

In this work, we study the implementation of the call-back option in a time-varying environment with customer choice. We consider a queuing model with two types of queues: online and call-back, where customers choose their preferred queue based on information regarding the expected online wait and maximal offline wait. As opposed to most service centers nowadays which offer a constant delay guarantee, we propose to change the promised delay guarantee depending on the time of day. In particular, in overloaded periods, we will reduce the delay guarantee to make the call-back option more attractive. We used an iterative simulation-based optimization algorithm to find the desirable delay guarantee for each time interval. We then analyze the performance of the IDG algorithm through a case study of a medium-sized call center. The case study shows that the IDG manages to reduce online waiting times to the desired level and improve

various other system performance measures. We also conduct extensive sensitivity analyses by comparing the IDG algorithm performance to other delay setting methods. We find that when resources become limited or fluctuate, the performance of IDG exceeds all other methods. Furthermore, our method can guarantee that the online waiting time won't exceed a predefined threshold and checks whether the promised delay guarantee is even feasible.

REFERENCES

Armony, M., and C. Maglaras. 2004a. "Contact Centers with a Call-Back Option and Real-Time Delay Information". *Operations Research* 52(4):527–545.

Armony, M., and C. Maglaras. 2004b. "On Customer Contact Centers with a Call-Back Option: Customer Decisions, Routing Rules, and System Design". *Operations Research* 52(2):271–292.

Ata, B., and X. Peng. 2017. "Managing the Callback Option under Arrival Rate Uncertainty". https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2947368.

Dong, J., E. Yom-Tov, and G. B. Yom-Tov. 2018. "The Impact of Delay Announcements on Hospital Network Coordination and Waiting Times". *Management Science*. https://doi.org/10.1287/mnsc.2018.3048.

Feldman, Z., A. Mandelbaum, W. Massey, and W. Whitt. 2008. "Staffing of Time-Varying Queues to Achieve Time-Stable Performance". *Management Science* 54(2):324–338.

Ibrahim, R., and W. Whitt. 2011. "Wait-Time Predictors for Customer Service Systems with Time-Varying Demand and Capacity". *Operations Research* 59(5):1106–1118.

Legros, B., O. Jouini, and G. Koole. 2016. "Optimal Scheduling in Call Centers with a Callback Option". *Performance Evaluation* 95:1–40.

Mandelbaum, A., and S. Zeltyn. 2004. "The Impact of Customers' Patience on Delay and Abandonment: Some Empirically-Driven Experiments with the M/M/N+G Queue". *Operations Research Spectrum* 26(3):377–411.

Senderovich, A., M. Weidlich, A. Gal, and A. Mandelbaum. 2014. "Queue Mining – Predicting Delays in Service Processes". In *Advanced Information Systems Engineering. CAiSE 2014*, edited by M. Jarke et al., Volume 8484 LNCS, 42–57: Springer, Cham.

Yom-Tov, G. B., and A. Mandelbaum. 2014. "Erlang-R: A Time-Varying Queue with Reentrant Customers, in Support of Healthcare Staffing". *Manufacturing & Service Operations Management* 16(2):283–299.

AUTHOR BIOGRAPHIES

GALIT B. TOM-TOV is an Assistant Professor at the Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Israel. She holds a Ph.D. in Operations Research and an M.Sc. in Industrial Engineering from the Technion. She was a Postdoc Fellow and an Adjunct Professor at the IEOR department, Columbia University, USA. Before starting her research career, she accumulated industrial experience by being a production engineer in Tambour and an information system specialist in PCB-Technologies. Her research interests include application of stochastic models and data analysis in Service and Healthcare Systems. She studies service system operations, both from a modeling/optimization perceptive and data analytics perspective. The latter is aimed at developing a better understanding of customer behavior and its operational implication. Her website address is https://gality.net.technion.ac.il and her e-mail address is gality@technion.ac.il.

TALI ZEITLER works as a business analyst in one of Israel's leading defense technology companies. She received her B.Sc. degree in Industrial Engineering an Management in 2014 and her M.Sc. degree in Industrial Engineering in 2017 both from the Technion—Israel Institute of Technology. Her research focused on service systems. Her e-mail address is tali.gertz@gmail.com.