Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

CLOSING THE SERVICE: CONTRASTING ACTIVITY-BASED AND TIME-BASED SYSTEMATIC CLOSURE POLICIES

Antonio Castellanos[1], Andrew Daw[2], Amy Ward[1], and Galit B. Yom-Tov[3]

[1]Booth School of Business, The University of Chicago, Chicago, IL, USA
[2]Marshall School of Business, University of Southern California, Los Angeles, CA, USA
[3]Faculty of Data and Decision Sciences, Technion—Israel Institute of Technology, Haifa, Israel

ABSTRACT

We examine different policies for systematic service closure in messaging service systems. The system is modeled as an $M/UHP/1$ queue, where service times follow a history-based Hawkes cluster process. We propose and examine stopping-time rules that balance between queue length and the probability of prematurely closing conversations. In a simulation study, we compare two families of systematic closure policies: the first relies on predictive information regarding service progress, i.e., the conversation's activity levels, while the second relies on elapsed time without activity. When restricted to static threshold policies, both families provide similar performance. However, when allowing the threshold to vary with the system state, activity-level policies outperform the inactive-time policies. Moreover, a large difference is observed between static and dynamic threshold policies. We therefore conclude that state-dependent (i.e., dynamic) activity-based policy is the most promising candidate to achieve optimal closure rules.

1 INTRODUCTION

Recent years have seen a tremendous growth in text-based services offered by an increasing number of companies across various platforms (e.g., WhatsApp, or even the company's own mobile app). This growth can be attributed in part to customer preferences (RingCentral 2012) and partially to the financial advantages of on-platform sales promotion (Tan et al. 2019). However, many of the operational policies for managing such contact centers are not well understood. One such policy is the closure policy of conversations: if the customer does not say goodbye, under what conditions should a service interaction be closed? Castellanos et al. (2022) reported that many companies establish a fixed time threshold, ranging from 2 minutes in fast-paced services to 2 hours in longer ones, allowing customers to have time to reply and at the same time limiting the customer's span of inactivity before conversations are automatically closed.

However, by connecting closure policies to routing policies, Daw et al. (2024) suggest that such a policy may not be optimal. Specifically, they prove that under the widely-used "lightest-load" routing policy, the closure policy should be based not on the time of inactivity, but instead on the space of the probability of future activity, where conversations are closed only if the predicted probability that the conversation has ended is sufficiently high. If the closure policy does not precisely control for this closure success probability, then Daw et al. (2024) shows that lightest-load routing cannot be optimal. While Daw et al. (2024) proposed a prediction method by fitting Hawkes processes to messaging data, they did not consider how the target probability itself should be determined and how this choice impacts the system at large.

In this paper we explore two families of closure policies: those based on activity level, which allow the lightest-load routing to be optimal, and those based on inactivity time, which are commonly used in practice. We expand on the work of Daw et al. (2024) by suggesting that the closure policy problem presents a new service reliability trade-off: when the closure threshold is too high, the service efficiency is compromised by prolonging customers' length of stay (LOS), while if the closure threshold is too low, the company risks angering customers by prematurely closing their conversations. To understand why this is the case, let us delve into the operational dynamics of text-based services.

A text-based service comprises a series of text messages exchanged between a customer and an agent. The agent can be a live agent, a bot, or some combination of both. With the quality improvements of large language models and the rise in agent costs, many service providers try to automate as much of the service dialog as possible — this is done mostly for the initiation and the conclusion parts of the conversation, since they are more standardized. Service conclusion is a common example of automation: a computerized system monitors customer activity, and if no activity is detected for a pre-determined period, the system will send a series of messages confirming that the customer does not need anything and close the conversation. We refer to such an intervention as an *systematic closure* policy.

The problem lies in deciding when to activate this procedure. If done too early or too frequently, it can be very annoying to the customer, thus reducing service quality. If done too late, the agent might appear busy while they have finished serving the assigned customer. This inactive customer keeps the agent in a "busy" state and prevents a queued customer from starting service (Tezcan and Zhang 2014). It's worth noting that even when service is not automated, live agents may not always be sure whether the customer is present, has concluded service, or has abandoned the interaction silently (i.e., without indicating that they have left) (Castellanos et al. 2022). Hence, systematic closure policies are implemented even in "simple" contact center platforms that have no bots for managing standard tasks.

In this paper, we aim to expand on these issues and examine the following: (1) What is the difference between static policies based on a time window in which the service is inactive (referred to as an *inactive time policy*), such as those reported above, and policies based on the probability of future activity (referred to as an *activity level policy*), as suggested by Daw et al. (2024)? We note that the differences may not only be in performance but also in implementation complexity. While an activity level policy requires the company to track each conversation's activity, the inactive time policy does not, making it easier to implement. However, we will show that the activity level policy is much easier to interpret in terms of premature closure probability, thus making it easier for managers to determine a specific level. (2) How much can dynamic policies outperform the static policies mentioned above for systematic closure? Naturally, a dynamic closure policy is better than a static policy, simply because it offers the company more flexibility to keep conversations open for longer periods when there are no customers in the queue or to close conversations earlier when there are too many customers waiting. The question is, to what extent do dynamic policies outperform static ones, and more importantly, is there a difference between time-based and level-based *dynamic* policies?

Given the inherent challenges in theoretical analysis of contact center systems and complex Hawkes process models (e.g., history-driven service durations that mean that standard queuing analysis techniques have to be modified), we employ simulation techniques to examine different threshold structures and explore the trade-off between premature closure and wait. We present several policies that perform well numerically and use simulation to analyze them and to draw insights about the characteristics of their strengths.

To the best of our knowledge, this paper is the first to formalize the study of closure policies by leveraging this history-dependent modeling framework. Nevertheless, the problem we study certainly has conceptual forebearers in the related literature. For instance, this service closure problem can be viewed as a new variant of the speed-quality tradeoff in services (Anand et al. 2011; Kostami and Rajagopalan 2014). That is, faster services close sooner but may be more likely to end prematurely, and that would create a low quality customer experience. This direction also connects to the literature on determining healthcare service duration in response to capacity shortages (Chan et al. 2014; Armony and Yom-Tov 2024). By comparison to prior works, the scope of our model is more micro- than macroscopic, which also distinguishes our framework from much of the previous stochastic modeling literature on contact centers (Luo and Zhang 2013; Tezcan and Zhang 2014). Specifically, we draw upon the model proposed by Daw et al. (2024), which is oriented and empirically calibrated at the level of intra-service time-stamps, rather than system-level queueing metrics. In practice, companies typically employ time-based policies or fit specific machine learning models to determine when a conversation should be closed. To our knowledge, this is the first paper that attempts to solve this service reliability problem using a stochastic model. Hence, this work expands the opportunity for managerial insight.

The remainder of this paper is organized as follows. In Section 2 we define the history-driven stochastic model of service, through which we formalize systematic closure as a stopping time. To prepare for the analysis to follow, in Section 3, we specifically discuss two focal families of stopping times: those based on thresholds for the duration of prior inactivity (§3.1.1 and §3.2.1), and those based on hitting times for the probability of future activity (§3.1.2 and §3.2.2). These policies are presented in static (§3.1) and dynamic (§3.2) variants. We compare the performance of these policies through simulation in Section 4, and the paper closes with discussion and conclusions in Section 5. The code for all the simulations can be found in the following [Gitub Repository].

2 MODEL FORMULATION AND SYSTEMATIC CLOSURE PRELIMINARIES

Following the observation from Daw et al. (2024) that Hawkes cluster models can effectively capture the dynamics of contact center conversations, we utilize a univariate Hawkes process (UHP) cluster to model the history-driven service interaction. Specifically, in this paper we use simulation to study a single-server service system where customers arrive according to a Poisson process with rate $\lambda > 0$ and where service durations are distributed according to the following *systematically-closed Hawkes cluster process*.

Definition 1 (Hawkes Service Model with Systematic Closure) With $A_0 = 0$ as the initial message in the service without loss of generality, let N_t be a point process for the number of messages sent up to time $t \geq 0$ (excluding the initial message), where this point process is driven by the intensity defined as

$$\mu_t = \sum_{i=0}^{N_t} \alpha e^{-\beta(t-A_i)}, \tag{1}$$

A_ℓ is the epoch for the ℓth message for all $\ell \in \mathbb{Z}_+$. That is,

$$\mathrm{P}\left(N_{t+s} - N_t = n \mid \mathscr{F}_t\right) = \begin{cases} \mu_t + o(s) & n = 1; \\ 1 - \mu_t + o(s) & n = 0; \\ o(s) & n > 1, \end{cases} \tag{2}$$

where $\mathscr{F}_t$ is the natural filtration of the stochastic process. We will refer to $\alpha > 0$ as the *instantaneous impact* on the intensity upon each new message, and we will let $\beta > \alpha$ be the corresponding *decay rate* of that impact. Finally, we will let the *systematic closure time*, τ, be defined as an almost surely finite stopping time (i.e., $\mathrm{P}(\tau < \infty) = 1$) that is adapted to the filtration $\mathscr{F}_t$.

The queueing model we study will associate one such stopping time with each customer as their end of service time. Specifically, we will analyze the $M/UHP/1$ queueing model, meaning a single server queue with homogeneous Poisson arrivals and service durations given by Definition 1. In this paper, we are interested in *the design of these stopping times*, meaning the design of the policy for the systematic closure of the stochastic process, and how it will impact the distribution of each customer's duration of service and, by extension, the performance of the service system overall as measured by wait and service success. Following standard Hawkes process stability conditions, we will assume that $\alpha < \beta$ so that the number of messages is guaranteed to be finite and that the intensity will converge to 0 almost surely as time tends to infinity. (Let us note, though, that the stability of the Hawkes cluster model is *not* the same concept as the stability of the queueing model, which we will discuss in more detail in Sections 3 and 4.)

To motivate both the general idea of systematic closure and the specific philosophies that it might embody, consider the following sample path in Figure 1. This plot shows the full sample path of the Hawkes cluster model, and it also shows two candidate closure policies. First, the blue curve is the value of the intensity, μ_t, across time, and the red dots mark the message timestamps in the service. This sample path contains 16 messages, and each of these points "excites" the process and increases the rate of future messages, as seen in the corresponding jumps within μ_t. If the last message announced the closure of the service (e.g., if the customer says "goodbye"), then the service would be closed when this final correspondence was sent at approximately time 85, which we will call the *natural closure time*. However, many customers leave the service without a farewell (Ascarza et al. 2018), and thus we need a systematic closure policy. Operating with the assumption that *no* messages indicate the end of service, the policies will instead close based on observable stopping conditions in the Hawkes cluster model.

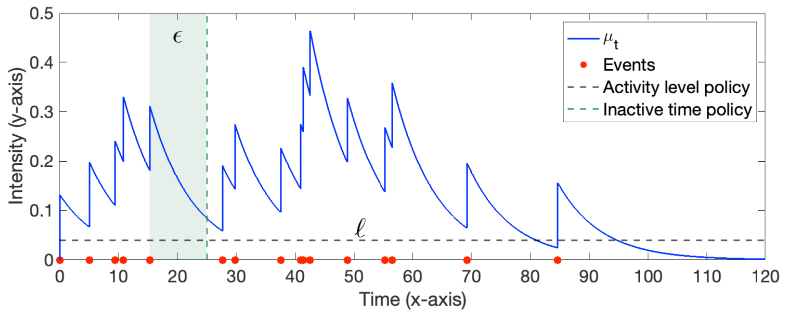


Figure 1: UHP sample path with $\alpha/\beta = 0.97$; $\beta = 8.15/60$. If the closure policy closes the conversation after $\varepsilon = 10$ time units of inactivity (green shaded area), the service will close at time $\tau \approx 25$ (green dashed line). If we use a closure level policy with level $\ell = 0.04$ (black dashed line), it will close at time $\tau \approx 81$.

Two such stopping times are demonstrated in Figure 1. First, let us consider a time-based policy like what is often employed in practice. For some $\varepsilon > 0$, this policy would close at the first moment in the service that ε time has elapsed without a new message, which occurs at time $\tau_\varepsilon = \inf\{t \geq \varepsilon \mid N_t - N_{t-\varepsilon} = 0\}$. As an example, the shaded green area in Figure 1 shows the first time that ε passes without a message, and the green dashed line shows when the service would be closed accordingly. Then, let us next consider a level-based policy. For some $\ell \in (0, \alpha)$, this policy would close at the first moment in the service that μ_t reaches ℓ, which occurs at time $\tau_\ell = \inf\{t \geq 0 \mid \mu_t = \ell\}$. The black dashed line shows the closure level, ℓ, and the service would end when this first intersects μ_t.

Both of these policies carry the risk that the systematic closure may be premature. Indeed, Figure 1 shows that each type of closure policy missed messages in this example. However, we can also observe that, although larger ε or lower ℓ could make it so that all messages occur in this example before systematic closure, this would come at the cost of wasted time. For example, $\varepsilon \approx 15$ would be longer than any gap in the sample path's message timestamps and thus would not close prematurely, but it would not do so until approximately time 100, meaning 15 time units after the natural closure. The story is very similar under level-based closure with $\ell \approx 0.02$: no messages would be missed, but the service would not end until μ_t reaches ℓ at approximately time 100. Notice that the natural closure time itself cannot be a systematic closure policy; because it is defined relative to future information and thus is not adapted to the filtration, it cannot be a stopping time.

Hence, both of these examples show the inherent tradeoff that managers face in designing systematic closure: a longer time (or a lower level) means that the policy is less likely to miss messages, but only at the cost of prolonged service and, by consequence, more waiting. Messaging service systems are designed to deliver services that are both reliable and readily available. Premature closure of services can undermine the system's dependability. Furthermore, no finite time nor positive level can guarantee that all messages would always be received. Because the gaps between timestamps are (dependent) continuous random variables, any systematic closure policy can at best offer a strong likelihood that the closure is not premature. Thus, this tradeoff presents a pair of salient performance metrics: the probability of premature closure and the expected number of waiting customers (or, by Little's Law, the mean waiting time) in steady-state.

Let us now recognize that the time-based and level-based policy families essentially approach this same tradeoff through the two different axes of Figure 1. First, the time-based policies are directly measured along the horizontal axis of Figure 1, as the green elements show. Then, by the Markovian nature of the UHP model, the probability that no messages remain in the service is a deterministic function of the current intensity (Laub et al. 2021):

$$\mathrm{P}\left(\lim_{s\to\infty} N_s - N_t = 0 \mid \mathscr{F}_t\right) = \mathrm{P}\left(\lim_{s\to\infty} N_s - N_t = 0 \mid \mu_t\right) = e^{-\mu_t/\beta}. \tag{3}$$

Hence, in some sense, the time-based policies reside on the horizontal axis, and the level-based policies are defined on the domain of the vertical. Throughout this paper, we will focus on this contrast, as the alignment of these policies allows us to highlight the tension of the tradeoff.

3 CLOSURE POLICES

In this section, we will further specify and inspect the two families of static policies we sketched in Section 2, and then we will introduce dynamic variants of each.

3.1 Static Policies

Following Section 2, we will study two static-threshold policies for terminating conversations based on certain conditions reached from the last message. The first policy employs a fixed closing *time* rule (§3.1.1), while the second utilizes a fixed closing *level* rule (§3.1.2). In each setting, we will discuss further properties of the policies, with a particular attention to the stability of the queueing system.

3.1.1 Inactive time policy

In a static time-based policy, a conversation is closed when ε time units have elapsed since the last message, which we have denoted as $\tau_\varepsilon = \inf\{t \geq \varepsilon \mid N_t - N_{t-\varepsilon} = 0\}$. Intuitively, this policy is straightforward to automate because the system only needs to track the time of the last message in each conversation. Moreover, the implementation is "model-free" and conveniently requires no other parameters.

In the model, though, it can be difficult to obtain closed-form expressions for the focal performance metrics or even for first-order statistics such as the mean service duration. This follows from the fact that the distributions of τ_ε and μ_{τ_ε} are not readily available. However, we can recognize two key facts. First, the service duration must be at least as long as the threshold time: $\tau_\varepsilon \geq \varepsilon$ almost surely by definition. Hence, it is possible that if ε is too large the system will not be stable: ε being less than $1/\lambda$ is necessary but not sufficient for the stability of the queue. On the other hand, for any arrival rate, there will exist an ε low enough such that the queue will be guaranteed to be stable. To recognize this, notice that (3) implies that the probability that there are no messages before ε time has elapsed since the start of service is $\mathrm{P}(\tau_\varepsilon = \varepsilon) = \mathrm{P}(N_\varepsilon = 0) = \exp(-\alpha(1-e^{-\beta\varepsilon})/\beta)$. Hence, $\mathrm{P}(\tau_\varepsilon = \varepsilon) \to 1$ as $\varepsilon \to 0$, meaning that the time of systematic closure will grow arbitrarily small as the threshold does. Notice, then, that these policies can be designed so that a service system that is unstable under the natural closure time of the true last message can actually be stable under systematic closure. However, this will require that a large fraction of services will be ended prematurely. Given the difficulty of analytically studying the way that the time-based policy manages this tradeoff, we will investigate this through simulation in Section 4.

3.1.2 Activity level policy

Our second proposed static policy closes a conversation when the correspondence rate intensity falls below ℓ, where ℓ is defined as some positive threshold below the initial level $\mu_0 = \alpha$. In Section 2, we denoted this as $\tau_\ell = \inf\{t \geq 0 \mid \mu_t = \ell\}$. By Definition 1, μ_t only decreases between the jumps at each message, and it does so continuously according to exponential decay at rate β. Hence, we could equivalently define this stopping time as $\tau_\ell = \inf\{t \geq 0 \mid \mu_t \leq \ell\}$, as the intensity will first cross ℓ whenever it first hits it.

Furthermore, by Lemma 1 of Daw et al. (2024), we know that level-based policies are the *only* systematic closure policies that both are almost surely finite and prescribe the closure probability exactly; any other finite stopping time would induce a non-trivial distribution across these closure probabilities. Following (3), we can see that by closing according to the proposed activity level policy, the successful closure probability will be precisely $e^{-\ell/\beta}$ (i.e., the premature closure probability is $1-e^{-\ell/\beta}$).

According to Proposition 1 below, one can also interpret τ_ℓ in terms of the expected number of messages in a conversation. That is, relative to the expected natural number of messages in the conversation, these level-based closure policies are expected to contain a fixed proportion of the overall messages.

Proposition 1 For some level $\ell \in (0,\alpha)$, let $\tau_\ell = \inf\{t \geq 0 \mid \mu_t = \ell\}$ be the associated level-based systematic closure time. Then, $\mathrm{E}[N_\tau] = (\beta-\ell)(\beta-\alpha)$, meaning $\mathrm{E}[N_\tau] = (1-l/\beta)\mathrm{E}[N]$.

Proof. First, let us notice that $N_t + \mu_t/(\beta-\alpha)$ is a martingale. From expressions for the mean of the Hawkes process (Laub et al. 2021), we can notice that the expectation of this quantity at time t, given its value at time 0, is $\mathrm{E}\left[N_t + \frac{\mu_t}{\beta-\alpha}\right] = N_0 + \frac{\mu_0}{\beta-\alpha}\left(1-e^{-(\beta-\alpha)t}\right) + \frac{\mu_0}{\beta-\alpha}e^{-(\beta-\alpha)t} = N_0 + \frac{\mu_0}{\beta-\alpha}$, and this verifies the claim. Then, by the optional stopping theorem (Daley et al. 2003, Theorem A3.4.VII) and the fact that the service model starts with $N_0 = 1$ and $\mu_0 = \alpha$, we have $\mathrm{E}[N_\tau] + \frac{\ell}{\beta-\alpha} = \mathrm{E}\left[N_\tau + \frac{\mu_\tau}{\beta-\alpha}\right] = N_0 + \frac{\mu_0}{\beta-\alpha} = 1 + \frac{\alpha}{\beta-\alpha}$, and this simplifies to the stated result. □

Implementing this policy requires the company to estimate α and β using historical data and track μ_t in real-time. However, by the Markovian property of the Hawkes process with exponential decay as employed in Definition 1, this can be achieved by simply maintaining a state for the value of μ_t at the last message (either before or after), along with the last message timestamp (Laub et al. 2021). For instance, for a state of the post-jump levels, the recursive calculation is $\mu_{A_i^+} = \mu_{A_{i-1}^+} e^{-\beta(A_i - A_{i-1})} + \alpha$. With α and β estimated from data, the company can thus obtain the ℓ that guarantees the desired closure probability.

Like the time-based policies, these level-based policies can also achieve stability for an otherwise unstable natural service duration. In a similar fashion, we can recognize that the time of deterministic decay from $\mu_0 = \alpha$ to ℓ is an almost sure lower bound on the systematic service duration: $\tau_\ell \geq \log(\alpha/\ell)/\beta$. Then, the probability that this lower bound is tight is $\mathrm{P}(\tau_\ell = \log(\alpha/\ell)/\beta) = \exp(-(\alpha-\ell)/\beta)$. Thus, as $\ell \to \alpha$, this probability likewise tends to 1. Hence, like the time-based policies, both artificial instability and artificial stability are possible if the level is too low or too high, respectively.

3.2 Dynamic Policies

Let us now propose six dynamic policies, which, like the static policies, terminate conversations based on either some time of inactivity or some activity level target. However, in the dynamic setting, we will now allow these conditions to vary depending on the state of the queueing system. In an attempt to find the best dynamic policy, we will examine several candidates of state-dependent functions for each type of rule.

3.2.1 Dynamic inactive time policy

With $q \in \mathbb{N}$ as the (known) number of customers presently waiting, let us now consider policies in which the ongoing service will be closed after time $\varepsilon(q)$ has elapsed from the last message if no new message was written. Hence, this is again a time-based policy, but now we will allow $\varepsilon(q)$ to be updated at every change of the queue length (i.e., at arrival or service completion).

Generally, we would like the family of policies to be such that the queue-length dependence of dynamic level, $\varepsilon : \mathbb{N} \to \mathbb{R}_+$, has the following properties: no service is closed when there is no one waiting ($\lim_{q\to 0}\varepsilon(q) = \infty$), the maximum inactivity time decreases as the queue length increases ($\varepsilon(q) \geq \varepsilon(q+1)$), and any service would be closed as the queue becomes infinitely long ($\lim_{q\to\infty}\varepsilon(q) = 0$). Notice that, unlike the static time-based policy, this structure will imply that the queue should be stable under any stationary arrival rate. That is, whenever the queue length grows to large, $\varepsilon(q)$ will tend towards 0.

To observe the effects of this queue-length dependence in a variety of settings, we will consider the following triplet of functional forms: (i) linear: $\varepsilon(q) = \Delta/q$, (ii) quadratic: $\varepsilon(q) = \Delta/q^2$, and (iii) square root: $\varepsilon(q) = \Delta/\sqrt{q}$. The $\Delta > 0$ coefficient need not be chosen to be the same in each case. Notice that cases (ii) and (iii) allow us to respectively consider convex and concave queue length dependence.

3.2.2 Dynamic activity level policy

Similarly, we will also consider a family of policies in which $\ell(q)$ is updated at every change of the queue length. In a manner similar to the time-based policies, we consider policies in which the queue-length dependence of dynamic level, $\ell : \mathbb{N} \to \mathbb{R}_+$, has the following properties: no service is closed when there is no one waiting ($\ell(0) = 0$), the closure level increases as the queue length increases ($\ell(q) \leq \ell(q+1)$), and any service would be closed as the queue becomes infinitely long ($\lim_{q\to\infty} \ell(q) = \infty$). We can now again notice that, because the closure level rises arbitrarily high with the queue length, this family of dynamic level-based policies will also guarantee stability of the queue for any stationary arrival rate.

To form a direct comparison with the time-based policies, we will also examine three versions of the level-based policy: (i) linear: $\ell(q) = \theta q$, (ii) quadratic: $\ell(q) = \theta q^2$, and (iii) square root: $\ell(q) = \theta\sqrt{q}$, where the $\theta > 0$ coefficient need not be chosen to be the same in each case.

Remark 1 The linear policy, $\ell(q) = \theta q$, can also be motivated through a notion of *virtual abandonment*. That is, if we artificially supposed that customers may abandon the service system, and that each patience time follows an independent and identical exponential distribution, we exactly obtain the linear closure policy. Specifically, we can define the following pseudo-objective function $f(\ell) = e^{-\theta\cdot q\cdot\log(\alpha/\ell)/\beta} \cdot e^{-\ell/\beta}$. Given the system has q customers waiting and closes conversations at level ℓ, the probability of successful closure is $e^{-\ell/\beta}$, and $e^{-\theta\cdot q\cdot\log(\alpha/\ell)/\beta}$ is the probability of no virtual abandonment happening from time 0 to the time that ℓ would be reached if no messages occur first. We thus wish to maximize $f(\ell)$. If we solve for the optimal ℓ given system parameters (μ_t, β, θ), we get that $\ell = \theta q$. In addition to reproducing the linear policy we have defined, it is interesting to observe that although the pseudo-objective, $f(\ell)$, allows for dependence on μ_t, this does not appear in the optimal level. Hence, the optimal virtual abandonment policy updates at every queue length change, but it does not change upon each intra-service message.

4 EXPLORING POLICY PERFORMANCE USING SIMULATION

As discussed in Section 2, we examine two measures of performance motivated by the core tradeoff inherent to systematic closure: the probability of premature closure and the average number of customers waiting. Because of the complexity of the models, we approach this investigation through simulation. Let us now outline the respective sampling methodologies we employ for the static and dynamic policy settings.

To obtain the steady-state values of these performance measures when using the static policies, we can simply simulate service durations in a vacuum. For each customer, we generate one sample path of the Hawkes cluster model with corresponding closure stopping time, and we then count the number of conversations that are prematurely closed to compute the probability. For the mean number of customers waiting, we leverage the Pollaczek–Khinchine (PK) formula for an M/G/1 queue (Khintchine 1932), and with the simulated conversations we obtain two of its required components, mean and variance of the service times under each stopping time rule. To simulate the UHP message timestamps, A_i, and calculate its intensity rate at time t, μ_t, we follow the approach outlined by Dassios and Zhao (2013), yet modify it to account for our systematic closure policies. The simulation works in the following way: at time t we generate two event times, S_1 and S_2, where the subscript 1 represents a new message and the subscript 2 a service closure. New message event time $S_1 = -\log(D)/\beta$, where $D = 1 + \beta\log(U_1)/\mu_t$, and $U_1 \sim \mathsf{Uni}(0,1)$, if $D > 0$, and $S_1 = \infty$, otherwise. This is based directly on the Dassios and Zhao (2013) procedure, where the random variable D manages the fact that S_1 has a degenerate distribution, in that (3) shows that it has a positive probability of being equal to infinity. If and only if $D > 0$, then S_1 is finite and there is a next message. Determining the service closure event time, S_2, depends on the specific static policy. For the level policy, S_2 is set to $\log(\mu_t/\ell)/\beta$ and for the time policy $S_2 = \varepsilon - (t - A^-)$, where A^- is the most recent message time and $t \geq A^-$ is the current time. If $S_1 < S_2$, then the next event of the conversation will be a new message being sent, and the current time and the intensity will be adjusted to $t = t + S_1$ and $\mu_{t+S_1} = \mu_t \cdot e^{-\beta S_1}$, respectively. But if $S_1 \geq S_2$, then the service will be closed and the time will be updated accordingly. Additionally, in the case that $D > 0$, the service will be prematurely closed (and counted as such). We simulate 2^{20} independent and identically distributed (i.i.d.) conversations for both static policies and for each parameter combination.

For dynamic policies, we cannot leverage the PK formula since the service durations are state-dependent, and thus the durations are not i.i.d. Therefore, we perform a full system simulation of the M/UHP/1 queue. Specifically, at time t we generate three event times S_0, S_1 and S_2, where subscript 0 is arrival, 1 is new message, and 2 service closure. S_0 is an interarrival time of a Poisson process with rate λ, $S_0 = -\log(U_0)/\lambda$ for $U_0 \sim \mathsf{Uni}(0,1)$. Then, S_1 is drawn using the same procedure we used for the static policy. To draw S_2 in the level-based policies, we use the state-dependent closure function $\ell(q)$, such that $S_2 = (\log(\mu_t/\ell(q))/\beta)^+$; in the time-based policies, we determine the inactivity time according to the $\varepsilon(q)$ function, and set $S_2 = (\varepsilon(q) - t + A^-)^+$. As in the static vacuum-based simulation, the next event occurs at the minimum of these times, at which point we update the time, intensity, and queue length accordingly. The simulation finishes when we reach the simulation horizon T, which we set to be 2^{13} in our experiments. We run the described system simulation with 2^{14} repetitions for each parameter combination.

As explained, there is a trade-off between the two performance measures; as one increases, the other decreases. Therefore, we will explore a Pareto frontier of the performance measures, where the better policy is one with lower values on both measures. When simulating the static policies, we vary the value of the level or time threshold between all possible values such that the system remains stable. In all simulations, the arrival rate is set to $\lambda = 1$ without loss of generality. To ensure stability, the systematic closure policy should ensure that the average service time is smaller than one. Specifically, for the static level policy, for every (α,β) combination we perform a simple binary search to find the minimum ℓ that will keep an upper bound of the average service time smaller than $\lambda = 1$. For the static time policy, we simply increase ε with a fixed step size until the mean service time is close to one.

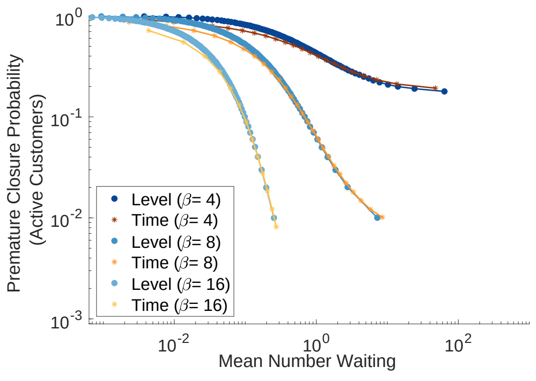
4.1 Comparing Static Policies

Figure 2 presents Pareto frontier graphs with the two performance measures for the static policies outlined in Section 3.1. The vertical axis represents the probability of premature closure among conversations that are not trivially inactive. That is, a trivially inactive conversation is defined as one that includes only the one initial message, as we assume that service requires at least one response. Therefore, by definition, a one-message conversation cannot be prematurely closed. As such, in the following figures, we normalize the premature closure probability by the probability of no responses to the initial message, $1 - e^{-\alpha/\beta}$. Then, the horizontal axis depicts the mean number of customers waiting in the queue. Because $\lambda = 1$, this is also equal to the mean waiting time per customer via Little's Law.

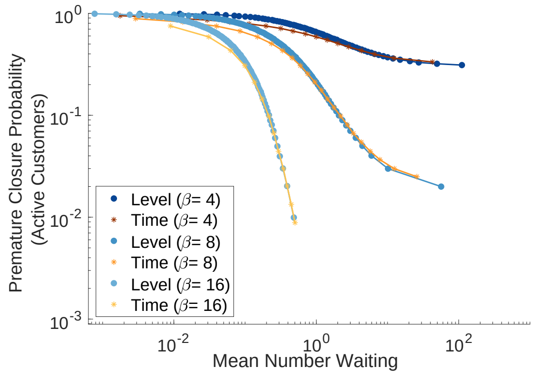
We vary the system conditions between subfigures by changing α and β, such that the average number of messages in a conversation until natural closure, $\mathrm{E}[N]$, varies across a range of service cluster sizes. In particular, we set α/β values to be in $\{0.8, 0.9, 0.95, 0.99\}$ so that $\mathrm{E}[N] = 1/(1-\alpha/\beta)$ ranges from 5 to 100. These represent upper bounds of the mean number of messages that we would get for each configuration; by Proposition 1 we know that this is directly proportional to the mean number of messages in the level-based policy. Each subfigure in Figure 2 corresponds to one combination. Within each subfigure, we choose three values of β ($\beta \in \{4, 8, 16\}$), so that the Hawkes process varies from a slow-paced process (low β) to a condensed or fast-paced process (high β). For example, changing β from 4 to 8 doubles the rate of messages while keeping the overall mean number of messages fixed. For the above parameter combinations, the natural service durations (calculated via Daw (2024)) and the corresponding queue lengths (assuming natural closure) are given in Table 1. As $\alpha/\beta \to 1$, the service becomes more active and, consequently longer. A long service duration can lead to system instability since the system utilization exceeds one.

Table 1: Natural closure duration and M/UHP/1 average queue length for each parameter combination.

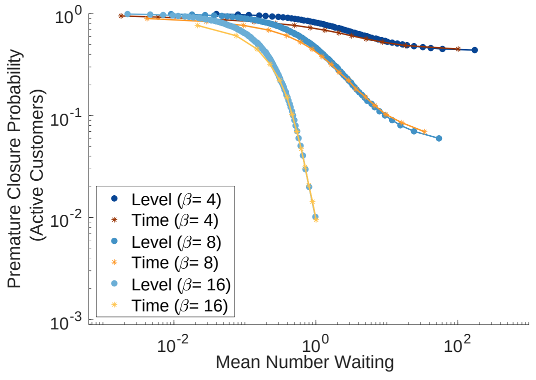
$\mathrm{E}[N]$	Natural closure - Avg. (STD)			Avg. queue length		
	$\beta = 4$	$\beta = 8$	$\beta = 16$	$\beta = 4$	$\beta = 8$	$\beta = 16$
5	0.91 (0.98)	0.54 (0.49)	0.31 (0.25)	9.83	0.58	0.12
10	1.24 (1.59)	0.71 (0.79)	0.40 (0.40)	∞	1.91	0.26
20	1.57 (2.45)	0.87 (1.22)	0.48 (0.61)	∞	8.51	0.58
100	2.35 (6.08)	1.26 (3.01)	0.67 (1.52)	∞	∞	4.21



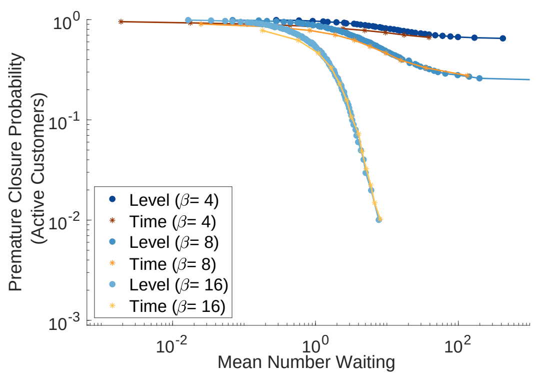
(a) $\mathrm{E}[N] = 5$ $(\alpha/\beta = 0.8)$



(b) $\mathrm{E}[N] = 10$ $(\alpha/\beta = 0.9)$



(c) $\mathrm{E}[N] = 20$ $(\alpha/\beta = 0.95)$



(d) $\mathrm{E}[N] = 100$ $(\alpha/\beta = 0.99)$

Figure 2: Pareto frontiers comparing static level and time policies (logarithmic scale).

From Figure 2, we observe that, in general, the differences between the two static policies are very small. The inactive time policy is slightly better if management is interested in keeping the queue low, while the activity level policy has a slight advantage when management is focused on reducing premature closures. The top left corner, where the premature closure probability is close to 1, corresponds to the smallest values of ε or the largest values of ℓ. In these cases, the company is the least patient with the customer and closes the service after a very short period of inactivity.

Lastly, regarding the influence of the UHP parameters on the performance, we observe that bigger β values allow us to push the stability bound (i.e., decrease the level ℓ and increase the time threshold ε) and therefore reduce the minimal achievable premature closure probability. This is because, as we can see in Table 1, increasing β values means increasing the speed of the service and reducing load.

4.2 Comparing Dynamic Policies

Figure 3 presents the Pareto frontier for the same two performance measures for the six dynamic policies presented in Section 3.2. We use the same parameter combinations as in Section 4.1.

Unlike the static case, here we observe a significant difference between the level- and time-based policies. It is clear that the level-based policies are much better, especially in moderate- or fast-paced services where β is moderate to high. We also observe that the level-based policies are particularly advantageous when one aims to create a balance between the two performance measures. Interestingly, the linear, quadratic, and square root dynamic level policies largely perform similarly. On the other hand, the dynamic time policies exhibit a significant difference between the linear, quadratic, and square root policies, where the quadratic outperforms the linear, which outperforms the square root policy. This is most apparent for moderate to low probabilities of premature closure.

When comparing Figure 3 to Figure 2, we observe the large advantage of dynamic policies over the static ones. (To facilitate the comparison, we include in Figure 3 the best performance achievable by a static policy under each β). The fact that there is a difference in itself is not surprising, as dynamic policies include the static ones by setting constant $\ell(q)$ and $\varepsilon(q)$. However, the magnitude of the difference is surprisingly large. The dynamic policies don't exceed more than 10–50% premature closure (depending on the system parameters) even with a long queue. This fact reveals a big difference between the static and dynamic policies. Static policies can achieve no queue only by setting the threshold so that conversations close almost immediately, while dynamic policies allow conversations to continue as long as there is no one waiting in the queue, reducing premature probabilities dramatically.

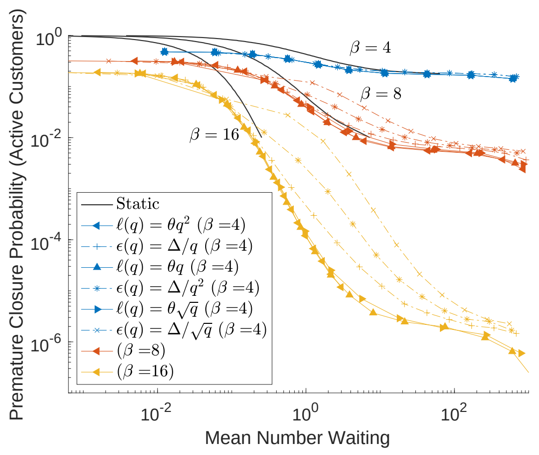
Next, we want to explore the influence of the UHP parameters on performance. When $\beta = 4$ the service is slow-paced relative to the arrival rate of 1. This means that the natural closure time is long and the system is overloaded (see Table 1). In this case, we observe that the closure probability has a low range of achievable values, while the queue length may vary from ≈ 0 to 10^3. The reason is twofold: first, in overloaded systems, there is little possibility of reducing conversation duration, and second, queue length is highly sensitive to that duration in high load conditions according to the PK formula. For this reason, when we examine the case of $\beta = 8$, where the service is twice as fast, we have higher flexibility to determine the policy threshold and can get a larger range of premature closure probabilities. When $\beta = 16$, the process is fast-paced, and the system remains stable even when the service duration is close to the natural closure time. Hence, we can achieve low premature closure probabilities.

5 DISCUSSION AND CONCLUSIONS

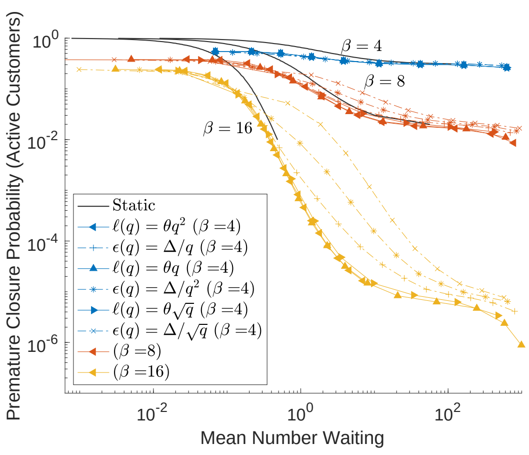
In this paper, we employed simulation to examine eight systematic service closure policies. Our findings indicate that the dynamic policies not only guarantee system stability, but they also outperform static policies. Both the time-based and level-based dynamic policies have their merits. If a company seeks low wait, a time-based policy might be suitable, albeit at the cost of a higher risk of prematurely ending services. If the company instead seeks to minimize the probability of premature closures (and therefore increase satisfaction), a level-based policy is better suited. Intuitively, the advantage of the level-based policies is that there is no degree of separation between the managerial decision of choosing a premature closure probability and choosing a level. To this end, this directness of control may explain why the magnitude of the level-based policy advantages exceeds that of the time-based policies, as seen in Figure 3.

In our search for best performing policies, we considered more than those presented here. This includes extensions such as extensions of the "virtual" abandonment policy in 3.2.2 (e.g., partial abandonments, different "virtual patience" distributions), and other pseudo-objective functions. However, these were not noticeably better than the presented policies, and were therefore not included due to space constraints. The optimal closure policy remains an open question that requires theoretical analysis.

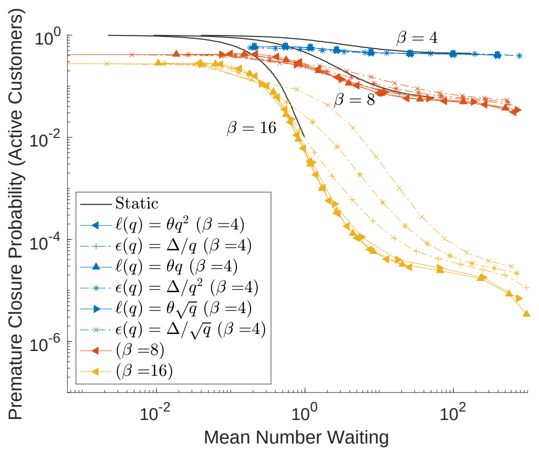
While this paper was motivated by contact center services, the systematic closure policy can be implemented in other contexts too. For example, premature service termination is referred to in healthcare



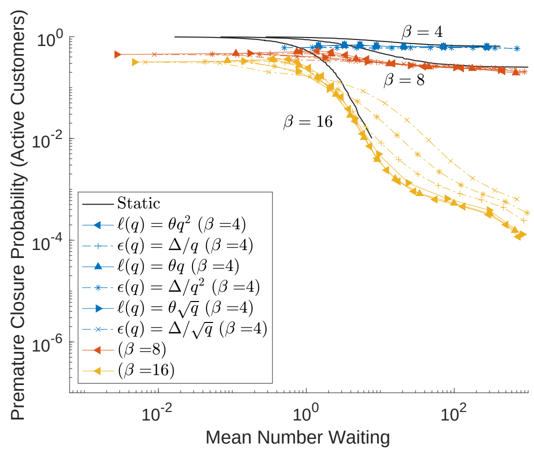
(a) $\mathrm{E}[N] = 5$ $(\alpha/\beta = 0.8)$



(b) $\mathrm{E}[N] = 10$ $(\alpha/\beta = 0.9)$



(c) $\mathrm{E}[N] = 20$ $(\alpha/\beta = 0.95)$



(d) $\mathrm{E}[N] = 100$ $(\alpha/\beta = 0.99)$

Figure 3: Pareto frontiers comparing dynamic level and time policies (logarithmic scale)

as service speedup, which was shown to reduce patient quality-of-care, and increase patient readmissions and mortality (Delasay et al. 2019, and references therein). Therefore, analyses of queues with service speedup were analyzed to increase efficiency and minimize risks (e.g., Chan et al. (2014)), but this literature has not considered history-dependent service models, nor the policies that arise from them.

Our simulation study concentrated on single-server systems, but we hypothesize that the advantages revealed here of the dynamic level-based policies will follow to multi-server systems, as well. Notably, in the multi-server setting, level-based policies retain the Markovian property, facilitating simulation. In contrast, time-based policies require tracking the last message across all conversations, which may complicate analysis at scale. However, it is not immediately obvious how one should design level-based or time-based policies in the multi-server setting, as it is not clear if, for instance, the same dependence on the queue length in the single server would remain appropriate. Similarly, it is not clear if the same time or level targets should be used in all parallel services. We look forward to pursuing these inquiries in future work.

ACKNOWLEDGMENTS

A. Daw and G. B. Yom-Tov are grateful for the support of this research by Grant No. 2022095 from the United States-Israel Binational Science Foundation (BSF).

REFERENCES

Anand, K. S., M. F. Paç, and S. Veeraraghavan. 2011. “Quality—Speed Conundrum: Trade-offs in Customer-Intensive Services”. *Management Science* 57(1):40–56.

Armony, M. and G. B. Yom-Tov. 2024. “Hospital versus Home Care: Trading off Pre- and Post-Discharge Infection and Mortality Risks”. Working paper, Technion.

Ascarza, E., O. Netzer, and B. G. Hardie. 2018. “Some customers would rather leave without saying goodbye”. *Marketing Science* 37(1):54–77.

Castellanos, A., G. B. Yom-Tov, and Y. Goldberg. 2022. “Silent Abandonment in Contact Centers: Estimating Customer Patience with Uncertain Data”. Technion working paper.

Chan, C. W., G. B. Yom-Tov, and G. Escobar. 2014. “When to Use Speedup: An Examination of Service Systems with Returns”. *Operations Research* 62(2):462–482.

Daley, D. J., D. Vere-Jones, *et al*. 2003. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer.

Dassios, A. and H. Zhao. 2013. “Exact Simulation of Hawkes Process with Exponentially Decaying Intensity”. *Electronic Communications in Probability* 18:1–13.

Daw, A. 2024. “Conditional uniformity and Hawkes processes”. *Mathematics of Operations Research* 49(1):40–57.

Daw, A., A. Castellanos, G. Yom-Tov, J. Pender and L. Gruendlinger. 2024. “The Co-production of Service: Modeling Services in Contact Centers Using Hawkes Processes”. Management Science (to appear).

Delasay, M., A. Ingolfsson, B. Kolfal, and K. Schultz. 2019. “Load Effect on Service Times”. *European Journal of Operational Research* 279:673–686.

Khintchine, A. 1932. “Mathematical Theory of a Stationary Queue”. *Matematicheskii Sbornik* 39(4):73–84.

Kostami, V. and S. Rajagopalan. 2014. “Speed–quality trade-offs in a dynamic model”. *Manufacturing & Service Operations Management* 16(1):104–118.

Laub, P. J., Y. Lee, and T. Taimre. 2021. *The elements of Hawkes processes*. Springer.

Luo, J. and J. Zhang. 2013. “Staffing and control of instant messaging contact centers”. *Operations Research* 61(2):328–343.

RingCentral 2012, Dec. “Texting for Work on the Rise Per RingCentral Survey”. Press Release.

Tan, X. J., Y. Wang, and Y. Tan. 2019. “Impact of Live Chat on Purchase in Electronic Markets: The Moderating Role of Information Cues”. *Information Systems Research* 30(4):1248–1271.

Tezcan, T. and J. Zhang. 2014. “Routing and Staffing in Customer Service Chat Systems with Impatient Customers”. *Operations Research* 62(4):943–956.

AUTHOR BIOGRAPHIES

ANTONIO CASTELLANOS is a Postdoctoral Principal Researcher at the University of Chicago in the Booth School of Business, hosted by Prof. Amy Ward. His research is in the intersection of data science and service-system operations. He received his PhD from the Technion advised by Prof. Galit B. Yom-Tov. This year he will join the Hebrew University Business School as an Assistant Professor. Email address: antonio.castellanos@chicagobooth.edu, website: https://voices.uchicago.edu/antoniocastellanos/

ANDREW DAW is an Assistant Professor of Data Sciences and Operations (DSO) in the Marshall School of Business at the University of Southern California. He is interested in applied probability and stochastic modeling for service operations, particularly at the intersection of behavioral operations. Much of his recent work has involved the self-exciting Hawkes process, where he has used the history-dependent stochastic process to model behavior that depends on interactions, influences, and impulses. His email address is andrew.daw@usc.edu, and his website is https://faculty.marshall.usc.edu/Andrew-Daw/.

AMY WARD is a Professor of Operations Management at the University of Chicago in the Booth School of Business. Her main interest is in service operations management, and particularly, in the development and analysis of stochastic process models that are relevant for service operations management decisions. She utilizes tools from statistics, optimization, probability theory, queueing theory, game theory, decision analysis, and simulation. She currently serves as the Editor-in-Chief of *Operations Research*. Her email address is amy.ward@chicagobooth.edu, and her website is https://voices.uchicago.edu/amyward/

GALIT B. YOM-TOV is an Associate Professor in the Faculty of Data and Decision Sciences at the Technion—Israel Institute of Technology, and co-director of the SEELab (https://seelab.net.technion.ac.il). She studies combination of service science and behavioral operations, building models to understand people’s behavior in service systems and to incorporate these behaviors into operational models. She leads a multidisciplinary research approach combining Data Science and Stochastic Modeling. Her email address is gality@technion.ac.il, and her website is https://gality.net.technion.ac.il.