The Co-Production of Service: Modeling Service Times in Contact Centers Using Hawkes Processes

Andrew Daw[b], Antonio Castellanos[a], Galit B. Yom-Tov[a], Jamol Pender[c], Leor Gruendlinger[d]
[a]Technion—Israel Institute of Technology, [b]University of Southern California Marshall School of Business,
[c]Cornell University School of Operations Research and Information Engineering, [d]LivePerson Inc.

In customer support contact centers, a successful service interaction involves a messaging dialogue between a customer and an agent. Both parties depend on one another for information and problem solving, and this interaction defines a *co-produced* service process. In this paper, we propose, develop, and compare new stochastic models for service co-production in a contact center. A key observation is that the customer and agent's co-produced service has cross- and self-exciting dynamics within each conversation. The cross-excitation stems from the two parties responding to one another, and the self-excitation captures one party sending follow-ups to their own prior message. Hence, messages beget messages, and we capture this phenomenon by introducing Hawkes point process models of the conversational services. These models distinguish between the role of the customer and of the agent, reflect the service process's dynamic evolution over time based on its own history, and include additional behavioral and operational aspects, including the agent's number of simultaneous assignments and measures of the amount of information and sentiment each message contains.

To evaluate our service co-production models, we apply them to an industry contact center dataset containing nearly 5 million messages. We show that the Hawkes models better represent the service dynamics than do the classic Poisson and phase-type models. Indeed, we find that service interactions are characterized by strong agent-customer dependency and the centrality of the process's cross- and self-excitation attributes. Finally, we use the proposed models to improve upon routing algorithms used in contact centers. We show how an activity-based dynamic routing based on predicted information easily computed from our models can outperform well-known and widely used concurrency-based routing rules and substantially reduce customer waiting time, demonstrating how these history-dependent stochastic models can improve operational decision making in practice.

1. Introduction

Most service operations research assumes that service duration is some random variable. Only a few attempts have been made to partition service duration into its finer elements, and much of that work uses the tasks composing the service as its basic elements while viewing service as a part of a larger activity network (e.g., Mandelbaum and Reiman 1998). This is a good approach when tasks are clearly distinguished, for example, tasks in a loan approval process. But when considering a single conversation (i.e., a single service interaction in a customer support contact center), the tasks or phases composing the service are ambiguous and hard to define, even more so in real time. Instead, we take a different approach and build stochastic process models for the duration of the service interaction by capturing the collaborative communication structure between the customer and the service agent. This view draws its inspiration from the literature about service *co-production*, wherein the customer and the agent collaborate to produce the service. The fundamental role a customer plays in the service system is of course well known; early definitions of the service economy noted that "productivity in many service industries is dependent in part on the knowledge, experience, and motivation of the consumer" (Fuchs 1968). We suggest that the co-produced service process should be modeled as a two dimensional stochastic process that captures the coupled interaction between the customer and the agent. This enables us to take a refined view of the service process as it evolves within a single interaction, yielding a path-dependent metric for the current level of service activity within a given conversation. We also go beyond proposing this family of theoretical models of co-produced service; we test the suggested models on real contact center data, showing that capturing the roles and elements within a conversation is indeed important and valuable to accurately represent the service progression as well as for operational decision making.

Service time is affected not just by the tasks to be completed but also by operational and behavioral factors. An agent's response time has been shown to be impacted by system load (Kc and Terwiesch 2009), concurrency (or amount of multitasking) (Kc 2013), and customer-expressed sentiment (Altman et al. 2020). For a review of such effects on service duration, see Delasay et al. (2019). The customer's response time has also been shown to be impacted by system load (Ilk 2020). Of particular interest here, recent papers have shown that customers and agents impact each other's behavior simultaneously. For example, Altman et al. (2020) showed that the customer's expressed sentiment influences agent response times and vice versa; Ashtar et al. (2021) showed that the two sides' sentiments impact one another within a service conversation. This notion of mutual impact fits our perspective of a service process as involving two distinct roles—customer and agent—that need to cooperate with one another to successfully co-produce the service. Hence, the service models we propose incorporate the reciprocal effect that an agent and customer have on one another through the pace of their responses, showing for the first time that not only will an agent's response time impact the customer's response time, but that the customer has the same impact on the agent. Our models also include other behavioral aspects such as system load, the amount of information each party contributes, and the strength of the sentiment each party expresses.

Dong et al. (2015) showed that incorporating such factors within operational service models is important (see also Wu et al. 2019), and that these factors impact the optimal staffing level. We claim that providing an accurate dynamic model for how service progresses in real time may have a profound impact on the possible designs of new routing algorithms for service systems. Specifically, we suggest that our model can be used to predict agent workload in real time and thus can be used to more appropriately balance load between agents. Using a large-scale data-driven simulation, we demonstrate that incorporating our conversational service models into the routing algorithm can indeed improve the wait-for-service (*outer wait*) by over 20% and likewise improve the wait-during-service (*inner wait*) by over 6%, all without increasing staffing or otherwise allocating more resources. This result coincides with previous findings showing that predictive information can improve service system operations, as was shown for healthcare operations by Xu and Chan (2016). This also coincides with the idea that the customer-agent service interaction should be considered in operational models of service systems, as was previously suggested by Roels (2014), albeit from a strategic view of the service design and the division of labor.

1.1. The Context of Contact Center Operations

We will apply this co-production modeling concept to text-based service conversations offered by many modern customer support contact centers.[1] Currently, asynchronous communication channels that serve customers via chat or messaging applications are slowly replacing call (voice-based) channels as the preferred way for customers to communicate with companies. Indeed, a survey conducted by a cloud-based communications provider found that 78% of respondents preferred to text with a company rather than call them (RingCentral 2012). Contact centers have also been recognized as important platforms for reaching potential customers and promoting sales (Yom-Tov et al. 2020, Tan et al. 2019). Recently, data from contact centers has enabled researchers to observe detailed information about the conversational dependencies between customers and agents (Rafaeli et al. 2020). The amount of available information about service encounters in contact center data is much more detailed than what is available for call centers. For example, while call center data generally only includes information regarding when an interaction started and ended, for contact centers we can know exactly what was written, when, and by which party.

The contact center environment has some important, but not necessarily unique, features which have attracted the attention of operations research in recent years. For example, customers can abandon a contact center's queue silently (Castellanos et al. 2019), which is similar to the unobserved abandonments in "ticket queue" models (Xu et al. 2007, Jennings and Pender 2016) and the left-without-being-seen phenomena in emergency departments (Batt and Terwiesch 2015). Another important aspect of contact centers is that agents can serve more than one customer concurrently (Tezcan and Zhang 2014, van Leeuwaarden et al. 2017, Long et al. 2019). Again, such multitasking

[1] In this paper, we will use the term contact center to refer to customer support centers that offer service through digital text communication, and we will use call center to refer to one that offers service through verbal phone communication.

is also apparent in settings such as hospital operations (Kc 2013, Goes et al. 2018), court systems (Bray et al. 2016), and social welfare agencies (Campello et al. 2017). This concurrency results in one of the key distinguishing factors between text-based contact centers and call centers: serving multiple customers at once alters the pace of the service interaction, which then directly affects its length. This is because the communication is *asynchronous*, meaning that its messages need not occur in quick succession or simultaneously. By comparison to a synchronized conversation in an in-person service or in a call center, dialogues in contact centers may have prolonged periods of inactivity. This phenomenon is rooted in the digital nature of the communication. A customer could, for example, send a message to the agent, then step away from her phone or computer, and not immediately respond to the agent's next reply. Hence, her responses are not necessarily synchronized with the agent's. The level of this asynchronicity can vary with the communication platform. For example, a web-chat communication is typically only mildly asynchronous, as service durations take place on the order of 10 minutes (Castellanos et al. 2019), while an email communication may be highly asynchronous, as durations can span from hours to even weeks (Halpin and De Boeck 2013). By comparison, a fully synchronized service interaction in a call center typically only lasts a few minutes (Gans et al. 2010).

In this paper, we use data from a moderately asynchronous setting: message-based contact center service conversations of a telecommunications company. In this data, the service durations are on the order of minutes to hours; see Section 3.1 for detailed summary statistics. It is important that we note that this long conversation duration need not mean that the agent is actively serving the customer for the full duration. Indeed, in analyzing conversation data, we notice long periods of times in which the conversation is inactive or "on pause." For example, Figure 1 shows a sample path of seven conversations held by the same agent concurrently (each customer appears in a separate line in the graph). One can clearly see that, for some reason, the conversation with customer 1 is inactive between 17:17 and 18:43. In general, we find in our data that, on average, conversations are inactive for 69% of their duration. These inactivity periods include time intervals in which the agent is unavailable because she is either serving other customers or on break, as well as periods in which the customer is inactive for long times for her own reasons. Thus, there may be long stretches of the conversation in which no one writes anything.

In this way, asynchronous communication offers benefits to both the customer and the agent. On the customer side, for example, this structure allows the customer to take breaks during the conversation if she so desires. At any point, she can choose to temporarily leave the conversation, perhaps to gather information related to the service interaction or simply to pursue other activities. On the agent side, these gaps offer the representative the flexibility to assist several customers

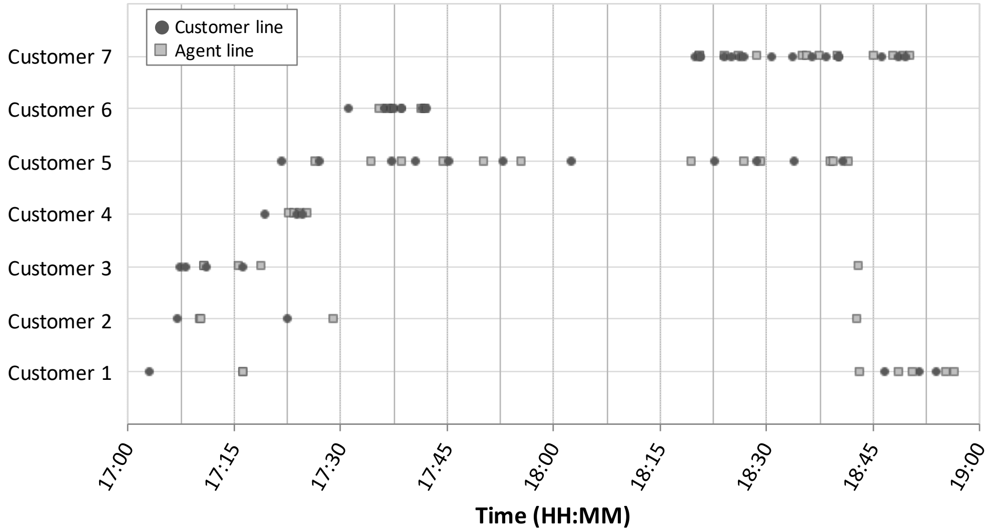


Figure 1 **Sample path of seven customers served by the same agent (May 4, 2017). Circles (squares) show when the customer (agent) sent a message.**

in parallel. Rather than idly waiting for one particular customer's response, an agent can instead make use of the spare time and assist another person.

While this flexibility is beneficial individually, it can be challenging for system-level decisions. For example, these behaviors create a mismatch between service time and conversation duration, and also between the number of active conversations an agent is handling and the number of conversations assigned to that agent—her concurrency (see Figure 2). These discrepancies can be problematic for operational decisions such as staffing and routing policies. Current contact center routing policies (e.g., Tezcan and Zhang 2014, Long et al. 2019) assume that all assigned customers are equally active at any given point in time. One of this paper's goals is to provide a methodology for measuring workload more accurately, aiming to evaluate, in real time, the true level of conversation activity that an agent is handling. Doing so has the potential for high practical impact on designing new operational models, as we will show.

1.2. Using Hawkes Processes to Model Service Dynamics

Taking a closer look at the dynamics depicted in Figure 1, we can observe that each conversation progresses in bursts of activity. Once one party sends a message (writes something and presses the "send" button), there is an increased chance that the other party will send a message too, leading to bursts of messages and a slight over-dispersion in the message arrival process ($CV = 1.07$). This could be related to the classical psychological concept of *foot-in-the-door* (Freedman and Fraser 1966), since when one party succeeds in engaging their partner in the co-production of service, the likelihood increases that the service will continue on. This also may be related to the

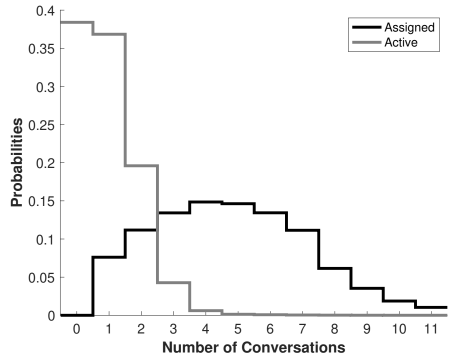


Figure 2 **Comparing the distribution of the number of active conversations an agent handles and the number of conversations assigned to an agent. The probability indicates the proportion of time an agent handles a specific number of conversations during her workday. May 22–31, 2017.**

notion of reciprocity, since when one party responds quickly, it may induce the other party to reciprocate and respond quickly too. We therefore propose that the service model should depend on the conversation history, i.e., the process sample path. This is different from classical service models that have traditionally been based on Jackson networks with an innate assumption of only immediate dependency between stages (e.g., Mandelbaum and Reiman 1998). These dynamics of service interaction bursts followed by inactive periods are reminiscent of a physical process such as an earthquake, where a sudden seismological event increases the probability of subsequent aftershocks. This type of stochastic behavior was classically modeled using a Hawkes process (HP), such as was done by Ogata (1988). Specifically, he showed that an earthquake can be viewed as a *self-exciting point process* in which each arrival "excites" the arrival rate, meaning it increases the probability of another arrival occurring soon afterwards. This stochastic intensity point process was originally defined by Hawkes (1971) and has been used to model contagion and virality in a wide variety of applications, such as financial markets (e.g., Embrechts et al. (2011), who adapt the process to daily stock market index data), social media (e.g., Rizoiu et al. (2017), who model retweet cascades), public health (e.g., Rizoiu et al. (2018), Daw and Pender (2021), who connect HPs to epidemic models), and queueing systems (e.g., Gao and Zhu (2018a), Daw and Pender (2018), Koops et al. (2018), Chen (2020), who study queueing systems with HP arrivals).

We claim that the framework the Hawkes process provides is appropriate for service-time modeling since it allows us to model interdependent events over continuous time and can be generalized to account for our desired service-encounter factors. A closely related stream of work uses Hawkes processes to model email communication (Halpin and De Boeck 2013, Fox et al. 2016). Here, HPs represent the dyadic communication structure between the two communicating entities; this serves as an important precursor for our work and constitutes an additional empirical justification for our modeling approach. That being said, there are considerable differences between our project and these two papers. For example, we study the effect of busyness-level features (e.g., concurrency of the agent) and message-level features (e.g., sentiment) on the conversational model, thus capturing both behavioral and operational elements that were not previously analyzed. A second difference stems from the information structure available in each environment—email data has a clear path of communication known through the Re and Fwd indicators, while in chat communication, no clear indication is given regarding which messages are related to each other. Indeed, in service conversations, it is common to see sequences of messages written by the same side (e.g., in Figure 1, the first two messages customer 3 wrote and the last two messages the agent wrote to customer 1). Another important difference relates to the second goal of our project: we aim to use this stochastic model of the service progression to inform operational decision making, and we demonstrate its usefulness by proposing a prediction-based routing algorithm. This gives our work a predictive analytics context within operational data science, as we want to use the history of a conversation to predict its future activity and then assign new customers to agents accordingly. Finally, by analyzing the coefficients of the Hawkes process conversation models in concept and through data, our work has a behavioral operations flavor, aiming to gain insight about the self-production vs. co-production nature of conversational service.

1.3. Contributions

We believe this paper contributes to two streams of literature: behavioral operations, by deepening the understanding of how service interactions evolve in reality, and service operations, by demonstrating the practical utility of our proposed stochastic models for managing conversational service interactions. We summarize these contributions here:

- We offer a new stochastic model for service encounters that is both history and relationship dependent. This model is based on a family of stochastic models called Hawkes processes, and it is flexible enough to incorporate additional operational and behavioral dependencies. We also prove a closed form stability condition for bivariate Hawkes processes in Theorem 1. This inequality offers an interpretable stability requirement for bivariate Hawkes processes in general and, in doing so, it simplifies the previous literature.

- We test our model's value on real data, comprised of nearly 5 million messages from a industry contact center. By comparing to both static and phase dependent (i.e., time-varying) service models, we show the significant benefits that result from capturing the history and role dependencies within co-produced service.

- Our Hawkes conversational models offer new insights regarding the mutual influence that agents and customers have on one another through their response times. In our case study, we show that customer response times impact agent response times and vice versa. When one party responds quickly, the other party is more likely to reciprocate by responding quickly as well. This reciprocal behavior has not been shown before. Moreover, each role has its own pace and dynamics. Agents tend to respond more quickly than customers and follow up on their own responses more frequently (i.e., are involved in more self-exciting behavior). The Hawkes models can also offer new insight regarding the strength with which each role drives the service interaction forward. Our case study reveals that, despite their individually slower pace, the customer is a much stronger driving force of the service progress across time.

- The Hawkes models also offer new insights regarding the history dependence of service conversations, showing that all the conversation's messages, not just the most recent one, impact customer and agent response times. This contradicts previous statistical models of service conversations, such as Altman et al. (2020), as well as Markovian representations of service networks, such as Mandelbaum and Reiman (1998), that a priori assume instantaneous effects.

- We demonstrate the operational importance of our model. Specifically, we show the model can be used to accurately measure each agent's dynamic workload, and in doing so, it can improve managerial decision making, such as in the routing of new customers to agents. In an entirely data-driven simulation experiment, we find that our models can improve well-known and widely used routing policies by projecting future conversation activity. We find that this can reduce the wait-for-service by over 20% and the wait-during-service by over 6%. On average across all the scenarios we consider, our models are able to reduce these waits by 11.5% and 4.1% respectively, all without allocation of additional resources or staff.

The remainder of our paper is structured as follows. In Section 2, we propose stochastic process models for the customer-agent service interaction. Specifically, we propose five models that successively build from treating the times between messages as independent and identical, to independent but non-identical, and to sample path and system dependent. In the same section, we prove the closed form stability condition for the Hawkes-based models and discuss its interpretations, provide key dynamic workload metrics, and briefly discuss computational details. Then, in Section 3, we calibrate the models using a large industry data set and evaluate their performance. In the case of the Hawkes models, we are also able to interpret the values of the estimated parameters and offer insight into the contact center service co-production. In Section 4, we use the estimated models for operational decision making in our data-driven simulation experiment, showcasing how these concepts can boost routing algorithms that have been popular both in practice and in the literature. We conclude with discussion in Section 5. All proofs are given in the appendix, as are complete computational details and other auxiliaries.

2. Modeling and Interpreting the Service Co-Production through Stochastic Processes

Our modeling goal in this work is to represent the service interaction as a stochastic process. In the context of contact centers, the service experience is the message-based communication between the customer and the agent. That is, their conversation is the service. It is where the customer's needs are both revealed and addressed, and where the agent's offerings are provided in response. Each passing message shapes the scope and direction of the service, and the customer and agent remained paired from the first message until the last. Our modeling philosophy will be to treat customer-agent pairs as homogeneous and allow the conversation they build to become unique as they correspond. This fits within our eventual operational aims of the model, which are to project future activity in a conversation given its history and then make managerial decisions accordingly. To begin, we will define our conversational service models.

2.1. Stochastic Process Models of the Conversational Service Duration

In conversational service, it always takes two to tango. The service occurs through a sequence of messages between the agent and customer, and the two parties work together to co-produce the service through their correspondence. As in any conversation, when one party asks a question or provides information, the other side is prompted to respond. Thus, the occurrence of each message makes it more likely that another message will occur soon after. Moreover, the pace of discussion and the turns to speak both depend on the history of the conversation thus far. In this section, we propose models of the service conversation that are not only path dependent, but also role and system dependent at their most general. Because the aforementioned characteristics match the hallmarks of self-exciting processes, our models are based on the Hawkes point process that introduced this concept into the stochastic process literature. We define three Hawkes-based models: the univariate (§2.1.2), the bivariate (§2.1.3), and the system bivariate (§2.1.4). Additionally, we will first define two models that more closely adhere to traditional models of service operations (§2.1.1). However, it is worth noting that all of the service duration models we consider will be stochastic processes, not just random variables. This is representative of our some of our goals in this work, as we aspire to capture the detailed richness of service dynamics. Furthermore, the increasing complexity of the five models will enable us to assess the impact of adding different behaviors and system dependencies to the conversational models. Specifically, in the Hawkes-based models, we endeavor to represent the dynamic nature of co-produced service, driven by the roles that the customers and agents play and by the collaboration they build through their conversations.

2.1.1. Static and Stage Dependent Conversational Models

Our most basic service models regard the service as a series of events, i.e., messages, that occur over time. Let A_i denote the time of the i^{th} message. Any service interaction stochastic process needs to define how those events depend on one another and when the service is expected to end.

As mentioned, classic service literature views the service process as a series of tasks, using a phase-type distribution as the stochastic model that captures the service progression (Mandelbaum and Reiman 1998). With this in mind, our most basic service model views the service as a repeating task Markov chain model, where the time between messages is exponentially distributed. This is similar in nature to repeating service models, such as the Erlang-R, used to describe service duration in hospitals and contact centers (see Yom-Tov and Mandelbaum 2014, Campello et al. 2017). In that model, the number of messages is geometric with mean $1/(1-p)$ and the service duration is viewed as the sum of i.i.d. exponentials generated via a phase-type model. Following this, our basic model assumes i.i.d. exponential response times with rate μ and a geometrically distributed number of messages, where all random variables are independent. We denote this model as the *sum of exponentials-static* (SES) model.

However, there is no reason to believe that the conversation phases progress with constant rate or satisfy a universal mean rate. Indeed, our data reveals that the number of messages in a conversation fits to a negative-binomial distribution rather than a geometric (see Figure 9) and that response time may vary with the conversation state—first increasing and then decreasing (see Figure 8). The latter was also confirmed by Altman et al. (2020) showing that the conversation's stage predicts agent response time. We cannot know in real time what stage the conversation has reached, but nevertheless, we can know how many messages the conversation has had so far. Therefore, our second service model generalizes the SES model by allowing for phase-varying rates for both the transition probability and the response time between messages. We call this model *sum of exponentials-dynamic* (SED). Here, the exponential random variables are independent but non-identical, and we draw the number of gaps from a negative-binomial distribution. In this way, both SES and SED may be viewed as Coxian phase-type constructions of the service duration, where SES has the same rate and absorption probability in every phase but SED allows the rates and probabilities to vary between phases.[2]

It is important to note, however, that both the SES and SED models are still path-independent. Although the SED rates may change from one phase to the next, the response time random variables are still independent from one another and from the total number of messages. So, while SES and SED may be well-represented in the literature, these stochastic processes cannot capture the

[2] We also considered gamma distribution variants of these models, i.e., SGS and SGD, but their performance was not different enough from SES and SED to merit their inclusion.

dependencies between messages. As we mentioned in the discussion of Figure 1, we observe strong dependency between messages and, moreover, that the service progresses with bursts of activity. Our next suggested model will capture that dynamic using self-exciting Hawkes processes.

2.1.2. History Dependent Conversational Model: Univariate Hawkes Process

Originally introduced in Hawkes (1971), the Hawkes process is a stochastic intensity point process in which the current intensity is determined by the history of events. In its Markovian form, this path-dependent process is represented as an intensity and counting process pair (ν_t, N_t) where N_t is the number of events occurring by time t, and where ν_t is given by

$$\nu_t = \lambda + (\nu_0 - \lambda)e^{-\beta t} + \int_0^t \alpha e^{-\beta(t-s)} \mathrm{d}N_s = \lambda + (\nu_0 - \lambda)e^{-\beta t} + \sum_{i=1}^{N_t} \alpha e^{-\beta(t-A_i)}, \tag{1}$$

with A_i as the i^{th} event epoch, $\lambda > 0$ as the baseline intensity, $\alpha > 0$ as the jump in intensity at each event, and $\beta > 0$ as the rate of decay in intensity between events. In this setting, the intensity is a univariate Markov process, and the intensity-counting process pair is a bivariate Markov process. Through this definition, each event excites the process by increasing the instantaneous rate of new events by α. To regulate itself, the intensity decays towards the baseline λ at rate β between events.

One of the most powerful perspectives on the Hawkes process comes from recognizing its branching process structure, originally identified in Hawkes and Oakes (1974). Because each event excites the process and contributes to the rate of new events, one can view the process in a parent-descendant fashion. That is, if one event causes excitement that spurs the occurrence of a subsequent event, the latter can be thought of as a descendant of the former. If an event is caused by the baseline rate, this can be thought of as an initial or baseline event. Then, every event is either a baseline event or a previous event's descendant. This allows us to observe the branches within the Hawkes process, as the progeny of baseline events form a family within the point process history. Within each branch, every event after the initial is by definition a descendant of a previous event in the branch. Perhaps most importantly, the branches are independent from one another, as the excitement caused by one event only affects its own branch. Similarly, this means that outside of the occurrence of the initial event, the branch does not depend on the baseline rate. That is, each branch only depends on its own history. A visualization of this is given in Figure 3.

This branching structure is exactly what makes the Hawkes process a natural model for the contact center service interaction. Each conversation constitutes a branch and each message an event. After the initial query, each message within a conversation is in response to some prior message in that conversation. For this reason, our modeling focus is on the ways branches function in the Hawkes process. To begin describing the various models we consider in this work, let us first detail the sequence of events in a messaging conversation. The customer always comes first,

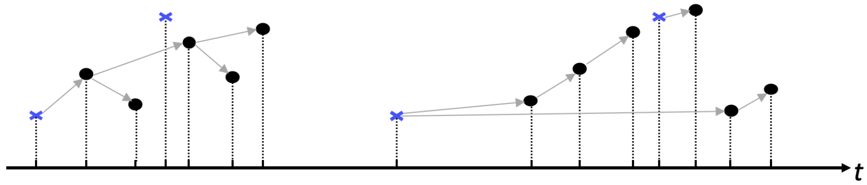


Figure 3 **An example of the Hawkes process branching structure, based on Laub et al. (2015). An x represents the starting point of a new branch, a circle represents an event that follows a previous event on an existing known branch, and an arrow represents the dependencies between events of the branch.**

literally. Every conversation starts with the customer's initial query; this constitutes the initial event on the branch. The customer and agent then correspond back and forth until the conversation ends, after which there are no more messages. It is important to note that this means each branch has finitely many descendants, since there are finitely many messages in every conversation.

We model the conversations as their own baseline-generated branches (i.e., we model each conversation independently), starting with a univariate model here and expanding to more complex Hawkes models in subsequent sections. Let us begin by defining the branches of a Hawkes process, derived from the full Hawkes process model in Equation (1). At time t in a conversation with N_t messages having occurred after the initial query at time 0, the rate of new messages will be given by the *univariate Hawkes process* (UHP) branch intensity is

$$\lambda_t = \alpha e^{-\beta t} + \sum_{i=1}^{N_t} \alpha e^{-\beta(t-A_i)} = \sum_{i=0}^{N_t} \alpha e^{-\beta(t-A_i)}, \tag{2}$$

where we explicitly define A_0 to be zero (representing the initial query). We can view the initial term in the middle expression in Equation (2), $\alpha e^{-\beta t}$, as a non-stationary baseline term that vanishes to zero at infinity. This is important because the message process should eventually stop, with no more messages being exchanged between the customer and agent. We can also view the UHP as a zero-baseline Hawkes process with an initial event at time 0 via the right-hand side of Equation (2). Again, in this case, it is clear that there will only be finitely many messages, as there is no outside source of new activity. In the messaging-based service context, we will refer to this branch intensity as the *correspondence rate*. Like before, A_i is the time of the i^{th} message after the initial query, α is the increase in the rate of new messages upon the arrival of a message, and β is the decay in the correspondence rate. Because there is no baseline intensity on the branches, the stability of the Hawkes process (assured when $\alpha < \beta$) implies there will only be finitely many descendants in any branch, matching our assumption of finitely many messages within any conversation. The closer α is to β, the more messages are expected in response to any given message, thus implying a longer and more active conversation. We establish and interpret formal stability results for all our Hawkes-based conversational models in Section 2.2.

We can also take a retrospective approach and reason about the influence of the process history within the Hawkes conversational model. Specifically, consider the relationship between the i^{th} message and the $(i-1)^{\text{th}}$ message. Given the process history up to the time of the i^{th} message, Poisson thinning arguments yield that we can calculate the probability that the i^{th} message is in response to its immediate predecessor through the ratio of the excitement from the $(i-1)^{\text{th}}$ message at time A_i, $\alpha e^{-\beta(A_i - A_{i-1})}$, and the overall correspondence rate just before the i^{th} message occurs, $\sum_{k=0}^{i-1} \alpha e^{-\beta(A_i - A_k)}$. By cancelling the $e^{-\beta(A_i - A_{i-1})}$ in the numerator and denominator, this is

$$\frac{\alpha e^{-\beta(A_i - A_{i-1})}}{\sum_{k=0}^{i-1} \alpha e^{-\beta(A_i - A_k)}} = \frac{\alpha}{\sum_{k=0}^{i-1} \alpha e^{-\beta(A_{i-1} - A_k)}} = \frac{\alpha}{\alpha + \sum_{k=0}^{i-2} \alpha e^{-\beta(A_{i-1} - A_k)}}, \tag{3}$$

where now the summation in the denominator of the rightmost expression is in fact the overall correspondence rate just before the $(i-1)^{\text{th}}$ message. While this expression could be further simplified by cancelling the α's, this form sheds insight. If the $(i-1)^{\text{th}}$ pre-message correspondence rate is greater than or equal to the message jump size α, then the i^{th} message is more likely to be in response to any message from 0 (the initial customer message) to $i-2$ than it is to be in response to message $i-1$:

$$\sum_{k=0}^{i-2} \alpha e^{-\beta(A_{i-1} - A_k)} \geq \alpha \iff \frac{\alpha}{\alpha + \sum_{k=0}^{i-2} \alpha e^{-\beta(A_{i-1} - A_k)}} \leq \frac{\alpha}{\alpha + \alpha} = \frac{1}{2}.$$

Hence, only if message $i-1$ occurs in a period of low activity should we expect the recent history to outweigh the past. We will further explore this concept and use such measures to inspect the level of history dependence in service conversations from true contact center data in Section 3.3.

2.1.3. History and Relationship Dependent Conversational Model: Bivariate Hawkes Process

The UHP model captures dependence on the history within a conversation, where an evolving collection of messages prompts new responses and thus increases the rate of new messages. However, it does not capture the full behavioral structure of this co-produced service because it conglomerates all messages into one stream. Thus, to reflect the different and dynamic behavior of the two separate parties in each contact center service interaction, we define the *bivariate Hawkes process* (BHP) conversational model, which consists of two different interacting intensities (i.e., two processes), one for each side of the conversation. When a jump occurs in one process, it increases both intensities, generating self-excitement and mutual excitement. In the messaging context, this means that there is an intensity for the customer message event process and an intensity for the agent message event process, and both intensities jump when either party sends a message. We will refer to these intensities as the customer and agent correspondence rates. Letting $\alpha^{\mathsf{c,a}}$ and $\alpha^{\mathsf{c,c}}$

be the jumps in the customer correspondence rate upon a new message being sent by the agent and by the customer, respectively, and letting $\beta^{\mathsf{c},\mathsf{a}}$ and $\beta^{\mathsf{c},\mathsf{c}}$ be the corresponding decay rates, the BHP customer correspondence rate is given by

$$\lambda_t^{\mathsf{c}} = \sum_{i=0}^{N_t^{\mathsf{c}}} \alpha^{\mathsf{c},\mathsf{c}} e^{-\beta^{\mathsf{c},\mathsf{c}}(t-A_i^{\mathsf{c}})} + \sum_{j=1}^{N_t^{\mathsf{a}}} \alpha^{\mathsf{c},\mathsf{a}} e^{-\beta^{\mathsf{c},\mathsf{a}}(t-A_j^{\mathsf{a}})}, \tag{4}$$

where N_t^{c} and N_t^{a} are the number of customer and agent messages, after the initial, up to time t, respectively, with A_i^{c} and A_j^{a} as the corresponding message times. Note that we now denote the initial customer message that begins the communication as $A_0^{\mathsf{c}} = 0$. Similarly, the agent correspondence rate is then analogously defined

$$\lambda_t^{\mathsf{a}} = \sum_{i=0}^{N_t^{\mathsf{c}}} \alpha^{\mathsf{a},\mathsf{c}} e^{-\beta^{\mathsf{a},\mathsf{c}}(t-A_i^{\mathsf{c}})} + \sum_{j=1}^{N_t^{\mathsf{a}}} \alpha^{\mathsf{a},\mathsf{a}} e^{-\beta^{\mathsf{a},\mathsf{a}}(t-A_j^{\mathsf{a}})}. \tag{5}$$

One important observation is that the various intensity jumps and decays need not be equal, thus conversations can involve both back-and-forth responses between parties as well as follow-up correspondence from one side alone, which we observe in reality. In this way, the conversation model allows for different influences between different types of messages so that, for example, a customer's message may be more likely to evoke a response message from the agent than from the customer again. This type of dynamic of both self- and mutually excited messages (i.e., both within and between roles) is clearly observed in Figure 1.

Before proceeding to our final and most general model, let us note that this BHP conversational model is itself a contribution to the literature of stochastic behavioral models of service interactions. Not only does this model capture the fact that there are two co-dependent sides in any service, it also reflects the fact that every service is a dynamic process shaped by what each party contributes. That is, Equations (4) and (5) show that while every conversation starts at the same place, $\lambda_0^{\mathsf{c}} = \alpha^{\mathsf{c},\mathsf{c}}$ and $\lambda_0^{\mathsf{a}} = \alpha^{\mathsf{a},\mathsf{c}}$, what follows is a continuous time and continuous state stochastic process. As the customer and agent write messages to one another and bring their own thoughts and requests to the table, the course of the service interaction evolves according to the pace the two sides set.

Because there are two roles in this service model, customer and agent, this leads to four relationships. We can see this reflected in the parameters: customer-to-agent, agent-to-customer, customer-to-self, and agent-to-self. Like in the UHP model, the α's increase the correspondence rates upon each message's receipt and the β's regulate these increases in the interim. The larger $\alpha^{i,j}$ is, the faster the pace in the i-to-j relationship, where $i, j \in \{\mathsf{c}, \mathsf{a}\}$. Furthermore, having $\alpha^{i,j}$ close to $\beta^{i,j}$ shows that the i-to-j relationship has high activity across time. For $i \neq j$, this shows strong responsiveness of one party to another, and for $i = j$, this shows significant rates of follow-up to one's own messages. Hence $i \neq j$ captures the co-production relationships of the service, and $i = j$

captures the self-production. In comparing the co-production relationships, these model parameters can reveal which party truly drives the conversation's progress. For example, if we find that $\alpha^{\mathsf{c,a}}/\beta^{\mathsf{c,a}} > (<)\, \alpha^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$ we can claim that the agent (customer) is the driving force of the service. That is, $\alpha^{\mathsf{c,a}}/\beta^{\mathsf{c,a}} > \alpha^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$ implies that a message from the agent invokes more response activity from the customer than a message from the customer would get from the agent. The analogous insight can be said for the customer if $\alpha^{\mathsf{c,a}}/\beta^{\mathsf{c,a}} < \alpha^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$ instead. We will demonstrate the impact of such calculations in our industry data case study in Section 4.

2.1.4. History, Relationship, and Content Dependent Conversational Model: System Bivariate Hawkes Process

The final model we propose incorporates additional behavioral and operational features into the service duration. We draw inspiration for use of these features from previous investigations in the literature. Because an agent can serve multiple customers in parallel, inner wait may occur due to the agent answering to other customers (Tezcan and Zhang 2014) and from the cognitive load on the agent due to multitasking during the focal conversation (Kc 2013, Bray et al. 2016). Therefore, we expect that when the agent handles more customers, the agent response time to each customer message should increase. This observation was empirically validated by Altman et al. (2020) in our context. Therefore, the agent's current number of simultaneous conversations, their concurrency, should be incorporated into the model. However, we can also note that concurrency should not impact the customer response time directly because the customer is not necessarily aware that the agent is serving other customers as well. Thus, concurrency will not be directly incorporated into the customer message intensity, only indirectly through the agent message intensity.

Because the service is co-produced by both parties, message context can impact the dialogue's future. Recent evidence from contact centers suggests the following two message-level features: (a) the amount of information each party contributes to the conversation in each message, and (b) the sentiment that is expressed in each message. Point (a) could be captured by the number of words in the message. When one party writes a long message, it gives the other party to which to respond, driving up the correspondence rate. Point (b) can be captured using sentiment analysis engines such as CustSent (Yom-Tov et al. 2018) that automatically measure the valence (positive or negative) and intensity of sentiment expressed in the text. Altman et al. (2020) showed that customer sentiment has an instantaneous influence on agent response time and effort and, therefore, total conversation duration. These two message-level features address the behavioral influences that the conversation's information can have on both the customer and the agent. Hence, we will incorporate the number of words and the sentiment into both the customer and agent message intensities. One can also note that, using only metadata features, our model is privacy-preserving, meaning that it does not directly use the text written within the conversations.

To add these features inside the Hawkes process models, let us introduce notation for the concurrency, the sentiment scores, and the number of words. Let K_t be the concurrency of the agent at time t, meaning the number of active conversations assigned to the agent at that time. The concurrency is a piecewise constant function of time, and we will assume that it is deterministically known up to the current time throughout the conversation. We will also assume that $K_t \in \{1, \ldots, \kappa\}$ for some $\kappa \in \mathbb{Z}^+$, representing the fact that the standard operational practice is to limit the maximum number of customers simultaneously assigned to one agent. Let S_i^{c} and W_i^{c} (S_j^{a} and W_j^{a}) be the sentiment score and word count, respectively, of the i^{th} (j^{th}) message from the customer (agent). We will also let the initial query sentiment and word count be S_0^{c} and W_0^{c}. We will assume that each of these four are drawn from four separate sequences of random variables that are independent from one another and from the conversational stochastic process. Each of the four sequences are individually identical in distribution. We note that Hawkes processes cannot have negative jumps; in order to ensure that, we normalize each sequence to be positive with unit mean. Altman et al. (2020) showed that negative emotions lead to more messages and longer conversations, therefore we also reverse-map the sentiment variable so that the larger S is, the more negative emotion is expressed. Thus, $S = 1$ corresponds to a message with neutral sentiment, and $W = 1$ corresponds to a message with an average number of words. One should note that these definitions show that we are using two different types of information. The concurrency uses the state of the contact center at time t and thus changes as time progresses, whereas the sentiment and word count are fixed with each message.

With these definitions in hand, we can now introduce the *system bivariate Hawkes process* (SysBHP) model. We define the SysBHP as having the customer correspondence rate given by

$$\lambda_t^{\mathsf{c}} = \sum_{i=0}^{N_t^{\mathsf{c}}} (\alpha_1^{\mathsf{c,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{c,c}} W_i^{\mathsf{c}}) e^{-\beta^{\mathsf{c,c}}(t - A_i^{\mathsf{c}})} + \sum_{j=1}^{N_t^{\mathsf{a}}} (\alpha_1^{\mathsf{c,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{c,a}} W_j^{\mathsf{a}}) e^{-\beta^{\mathsf{c,a}}(t - A_j^{\mathsf{a}})}, \tag{6}$$

and the agent correspondence rate given by

$$\lambda_t^{\mathsf{a}} = \sum_{i=0}^{N_t^{\mathsf{c}}} \frac{\alpha_1^{\mathsf{a,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{a,c}} W_i^{\mathsf{c}}}{K_t} e^{-\frac{\beta^{\mathsf{a,c}}(t - A_i^{\mathsf{c}})}{K_t}} + \sum_{j=1}^{N_t^{\mathsf{a}}} \frac{\alpha_1^{\mathsf{a,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{a,a}} W_j^{\mathsf{a}}}{K_t} e^{-\frac{\beta^{\mathsf{a,a}}(t - A_j^{\mathsf{a}})}{K_t}}. \tag{7}$$

Since this is our final and most fully featured model, let us briefly highlight the roles the different components play:

- Since each α is an instantaneous increase in the correspondence rate, it is measured per unit of time.[3] Likewise, because each β is a continuous rate of decay, it is also measured per unit of time.

[3] Because the α's are multiplied by either the message's sentiment score or word count in the SysBHP, they are also measured relative to these behavioral factors.

- Because the concurrency K_t is applied to both the instantaneous jump sizes and the decay rates, we can see that it modulates the time scale of the agent's response process. That is, as K_t increases, the agent correspondence rate slows, but so does its rate of decay. As we will see in the following subsections, this means that the agent's concurrency hampers her pace, but it does not alter the expected number of messages that will be sent. Rather, it simply slows them down. This is similar in nature to the philosophy of processor-sharing queueing models (e.g., Borst et al. 2005), which have been used in many prior models of contact centers (e.g., Tezcan and Zhang 2014).
- While the exponential kernel may suggest that the SysBHP is a Markov process when tracking a proper number of subprocess correspondence rates, this is not the case. When the concurrency changes and the agent gains or loses a customer, the agent's instantaneous correspondence rate must be recalculated using the complete history of the conversation. By comparison, the process in the BHP model is Markovian for a state containing the correspondence rate of each of the four directions of the conversation. Thus, the SysBHP is Markovian for the analogously defined state vector while on intervals in which the concurrency does not change. The full details of this are employed in the proofs contained in Appendix A.
- The jump coefficients $\alpha_1^{i,j}$ and $\alpha_2^{i,j}$ are both multiplied by unit mean random variables, and they are also both regulated by the same decay rate $\beta^{i,j}$. Thanks to the normalization, this allows us to compare the impact of the sentiment and the word count within each relationship. If the two α's are similar in value, then the sentiments and word counts have similar impacts within that direction of the conversation. Conversely, if one α is larger than the other, this relationship will be more sensitive to the corresponding features of each message.

Comparing the α's to one another and to the β's brings us to wonder about the stability conditions for these models. It is both conceptually natural and empirically justified to assume that each conversation will contain only finitely many messages, but understanding the conditions under which this will be true is valuable for both the design and understanding of service operations. In the following subsection, we will give a simple closed form inequality implying stability of the SysBHP and show what it can reveal about the relationships in service co-production.

2.2. Establishing and Examining Stability of the Hawkes Conversational Models

To begin thinking about the stability of our Hawkes conversational models, let us first define useful shorthand notation. For each $i, j \in \{\mathsf{c}, \mathsf{a}\}$, let $\bar{\alpha}^{i,j} = \alpha_1^{i,j} + \alpha_2^{i,j}$. Because the sentiments and word counts are normalized to be sequences of unit mean random variables, $\bar{\alpha}^{\mathsf{c},\mathsf{c}}$, $\bar{\alpha}^{\mathsf{c},\mathsf{a}}$, $\bar{\alpha}^{\mathsf{a},\mathsf{c}}$, and $\bar{\alpha}^{\mathsf{a},\mathsf{a}}$ are the mean jump sizes of the SysBHP (unadjusted for the concurrency) for each relationship within the service co-production. In Theorem 1, we introduce a closed form condition for the SysBHP (and by extension, all our Hawkes models) to be stable, implying that the correspondence rates in every conversation will eventually converge to 0 and that the total number of messages in any conversation will be finite.

This condition is written in terms of the ratios of the mean instantaneous jump size, $\bar{\alpha}^{i,j}$, and the decay rate, $\beta^{i,j}$. Because this fraction contains the drivers of the excitation and regulation in the communication process in its numerator and denominator, respectively, we will refer to it as the responsiveness ratio. Taking the subprocess of the agent responding to the customer's initial message as a demonstrative example, the occurrences of these responses form a non-stationary Poisson process with instantaneous rate $((\alpha_1^{\mathsf{a,c}} S_0^{\mathsf{c}} + \alpha_2^{\mathsf{a,c}} W_0^{\mathsf{c}})/K_t)\, e^{-\beta^{\mathsf{a,c}} t/K_t}$ at time $t \geq 0$. Integrating across all time, the total number of messages the agent sends in response to the customer's initial message is Poisson distributed with mean $(\alpha_1^{\mathsf{a,c}} S_0^{\mathsf{c}} + \alpha_2^{\mathsf{a,c}} W_0^{\mathsf{c}})/\beta^{\mathsf{a,c}}$; in expectation relative to the sentiment and word count, this is the responsiveness ratio $\bar{\alpha}^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$. So, the larger the responsiveness ratio, the more active we should expect that direction of the conversation to be.

Theorem 1. *If*

$$\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} \frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} < \left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right) \left(1 - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right), \tag{8}$$

then the SysBHP is stable, and

$$\lim_{t\to\infty} \lambda_t^{\mathsf{c}} = 0, \qquad \lim_{t\to\infty} \lambda_t^{\mathsf{a}} = 0, \qquad \textit{and} \qquad \lim_{t\to\infty} N_t < \infty$$

almost surely.

To the best of our knowledge, Theorem 1 provides the first closed form stability condition for (marked) bivariate Hawkes processes with four different exponential decay rates and four different (mean) jump sizes. Not only can this simplify to the well-known univariate condition $\alpha < \beta$, it also generalizes previously stated stability conditions of bivariate Hawkes processes with more restrictive assumptions. For example, Equation (7) of Bacry et al. (2015) is equivalent to (8) above for a BHP model in which $\alpha^{\mathsf{c,c}}/\beta^{\mathsf{c,c}} = \alpha^{\mathsf{a,a}}/\beta^{\mathsf{a,a}}$ and $\alpha^{\mathsf{c,a}}/\beta^{\mathsf{c,a}} = \alpha^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$. Theorem 1, like the aforementioned results in the literature, is derived from a general Hawkes process stability condition based on the spectral radius of a matrix of excitation kernels. The simple form in Equation (8) is achieved for the BHP through manipulation of this matrix, and then extended to the SysBHP through Doob's martingale convergence theorem; the full details of the proof are given in the appendix. By comparison to the univariate requirement that $\alpha < \beta$, we can see that each of the self-production terms must satisfy this condition, i.e., $\bar{\alpha}^{\mathsf{c,c}} < \beta^{\mathsf{c,c}}$ and $\bar{\alpha}^{\mathsf{a,a}} < \beta^{\mathsf{a,a}}$, since all quantities are positive, but it is possible that the (Sys)BHP is stable with one of $\bar{\alpha}^{\mathsf{c,a}}/\beta^{\mathsf{c,a}}$ and $\bar{\alpha}^{\mathsf{a,c}}/\beta^{\mathsf{a,c}}$ larger than 1.

In addition to these contributions to the Hawkes process literature, interpreting Theorem 1 provides insight into the behavioral structure and dynamics within co-produced service. Examining

Equation (8), the left-hand side of the inequality contains the co-production terms. These are the responsiveness ratios for each party responding to the other: agent-to-customer and customer-to-agent. On the other hand, the right-hand side is the product of the complements of the self-production ratios. The inequality connecting these products reveals many aspects of these relationships. For example, if the left-hand side is close to 1, the self-production terms must both be close to 0: this means that the dominant service structure is in the co-dependence and collaboration of the customer and the agent. Each party relies on the other to achieve the service goals, and the service encounter is driven by the interaction between them. Conversely, if the left-hand side is close to 0, then at least one directional relationship (i.e. agent-to-customer or customer-to-agent) is not very strong, and the service does not heavily depend on the two parties' interaction. In this case, both self-production terms may be close to 1, reflecting a service in which the two parties are loosely coupled—each side can largely work independently and complete tasks on their own. Moreover, if co-production is a key component of the service, this implies that we should expect the SysBHP and BHP models to be better fits than the UHP, as the two former models can feature different behaviors by each party: each side can have different paces, different word counts or sentiments, different reactions, and different dependencies on the system load.

2.3. Dynamic Workload Metrics of the Hawkes Conversational Models

In this section, we show how one can project a conversation's future activity based on its history. For any given conversation, the primary workload metric we consider will be the expected number of messages in an upcoming time interval. Traditional queueing theoretic notions of workload measure the remaining service times for all jobs in the system, computed by summing the residual work times across all arrivals to the queue so far. By comparison to the traditional notion of workload, the metric we use here is not a sum of the time remaining but rather a count of the number of messages remaining. Furthermore, here we (a) measure this for each conversation separately, (b) acknowledge that agent workload should sum only active conversations, and (c) use the dependence of the message timing on the process history (hence, the customer service time is endogenous rather than exogenous). In a sense, we are replacing the traditional notion of arrivals to the queue with the history of the conversation so far within the single service exchange, while also replacing the residual service time with the number of upcoming messages. We switch to using the number of messages because the time remaining in a contact center service may be misleading. Recall here our discussion of Figure 1 and the frequent conversation inactivity: it is possible that the end of the conversation is very far away but that there will also be no messages sent until much later from now. Hence, we instead calculate the expected number of messages in an upcoming interval to properly assess how busy a particular conversation (and cumulatively, a particular agent) will be over a time frame in which we seek to make an operational decision.

To compute these dynamic and history-based workload metrics, let us suppose that a conversation has been observed up to a time $t_0 \geq 0$. Let $\boldsymbol{\lambda}_{t_0} = [\lambda_{t_0}^{\mathsf{c,c}}, \lambda_{t_0}^{\mathsf{c,a}}, \lambda_{t_0}^{\mathsf{a,c}}/K_{t_0}, \lambda_{t_0}^{\mathsf{a,a}}/K_{t_0}]^{\mathrm{T}}$ be the vector of instantaneous correspondence rates at t_0, where these each encode the conversation's history so far in terms of the respective co-production and self-production relationships:

$$\lambda_{t_0}^{\mathsf{c,c}} = \sum_{i=0}^{N_{t_0}^{\mathsf{c}}} (\alpha_1^{\mathsf{c,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{c,c}} W_i^{\mathsf{c}}) e^{-\beta^{\mathsf{c,c}}(t_0 - A_i^{\mathsf{c}})}, \qquad \lambda_{t_0}^{\mathsf{c,a}} = \sum_{j=1}^{N_{t_0}^{\mathsf{a}}} (\alpha_1^{\mathsf{c,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{c,a}} W_j^{\mathsf{a}}) e^{-\beta^{\mathsf{c,a}}(t_0 - A_j^{\mathsf{a}})},$$

$$\frac{\lambda_{t_0}^{\mathsf{a,c}}}{K_{t_0}} = \sum_{i=0}^{N_{t_0}^{\mathsf{c}}} \frac{\alpha_1^{\mathsf{a,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{a,c}} W_i^{\mathsf{c}}}{K_{t_0}} e^{-\frac{\beta^{\mathsf{a,c}}(t_0 - A_i^{\mathsf{c}})}{K_{t_0}}}, \qquad \frac{\lambda_{t_0}^{\mathsf{a,a}}}{K_{t_0}} = \sum_{j=1}^{N_{t_0}^{\mathsf{a}}} \frac{\alpha_1^{\mathsf{a,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{a,a}} W_j^{\mathsf{a}}}{K_{t_0}} e^{-\frac{\beta^{\mathsf{a,a}}(t_0 - A_j^{\mathsf{a}})}{K_{t_0}}}. \tag{9}$$

As we have previously noted, these four values fully describe the dynamics of the probability model on intervals in which K_t is unchanged. Thus, given the current concurrency level and the four values in Equation 9, we can project the number of messages that will occur over an upcoming time frame based on what has occurred in the conversation so far. In Proposition 1, we give a matrix computation for this expected value.

Proposition 1. *Given $\boldsymbol{\lambda}_{t_0}$, the history of the system until time t_0, and assuming that the concurrency is constant from t_0 to time $t \geq t_0$, the expected number of upcoming messages sent from time t_0 until time t is*

$$\mathrm{E}\left[N_t - N_{t_0} \mid \boldsymbol{\lambda}_{t_0}\right] = -\mathbf{v}^{\mathrm{T}} \mathbf{M}^{-1} (\mathbf{I} - e^{\mathbf{M}(t - t_0)}) \boldsymbol{\lambda}_{t_0} \tag{10}$$

where $\mathbf{v}$ is an all-ones column vector and

$$\mathbf{M} = \begin{bmatrix} -(\beta^{\mathsf{c,c}} - \bar{\alpha}^{\mathsf{c,c}}) & \bar{\alpha}^{\mathsf{c,c}} & 0 & 0 \\ 0 & -\beta^{\mathsf{c,a}} & \bar{\alpha}^{\mathsf{c,a}} & \bar{\alpha}^{\mathsf{c,a}} \\ \frac{\bar{\alpha}^{\mathsf{a,c}}}{K_{t_0}} & \frac{\bar{\alpha}^{\mathsf{a,c}}}{K_{t_0}} & -\frac{\beta^{\mathsf{a,c}}}{K_{t_0}} & 0 \\ 0 & 0 & \frac{\bar{\alpha}^{\mathsf{a,a}}}{K_{t_0}} & -\frac{\beta^{\mathsf{a,a}} - \bar{\alpha}^{\mathsf{a,a}}}{K_{t_0}} \end{bmatrix}, \tag{11}$$

with $\boldsymbol{\lambda}_{t_0}$ as defined in Equation (9).

Proposition 1 gives us a way to predict a conversation's future level of activity based on its history. This allows us to assess the level of work in an upcoming interval of a conversation, which can then be used to distinguish the different expected workloads in each conversation or for each agent, since the number of messages corresponds to the number of tasks an agent must complete. This prediction is the basis of our new conversation routing policies, which we explore in Section 4. We can also compute these quantities for an infinite horizon. In Proposition 2, we give direct expressions for the total number of messages remaining in a conversation, both individually by each party and overall.

Proposition 2. *Excluding the total number of messages already sent in the observation period up to time $t_0 \geq 0$, the expected number of remaining messages the customer will send is*

$$\mathrm{E}\left[N_\infty^{\mathsf{c}} - N_{t_0}^{\mathsf{c}} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\left(1 - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)\left(\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} + \frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right) + \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\left(\frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} + \frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)}{\left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1 - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right) - \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}}, \tag{12}$$

and the total expected number of remaining agent messages is

$$\mathrm{E}\left[N_\infty^{\mathsf{a}} - N_{t_0}^{\mathsf{a}} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} + \frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right) + \left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(\frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} + \frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)}{\left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1 - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right) - \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}}, \tag{13}$$

hence the total expected number of messages from the present until the end of the conversation is

$$\mathrm{E}\left[N_\infty - N_{t_0} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\left(1 + \frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)\left(\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} + \frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right) + \left(1 + \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(\frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} + \frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)}{\left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1 - \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right) - \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}}, \tag{14}$$

where $\boldsymbol{\lambda}_{t_0}$ is as defined in Equation (9).

By comparison to the transient expressions in Proposition 1, we can see that, outside of the initial t_0 values, these long-range expectations do not depend on the concurrency. This hearkens back to our discussion on the time-modulation effect of the concurrency within the SysBHP. In the short term, the agent's number of assignments alters the agent's time scale, but across all time, these temporal effects do not impact the overall message count. This can also be observed in the distribution of the time until the next message, which we give in Proposition 3. There is a non-zero probability that no more messages occur, and again this long-run quantity does not depend on the concurrency beyond the initial condition. However, the probability that the time until the next message is merely no less than some finite value does depend on the way the concurrency alters the agent's decay rates.

Proposition 3. *Let S be the time until the next message occurs after time t_0. Assuming that the concurrency is constant from t_0 onward, the probability that no message is sent in the next x time units is*

$$\mathrm{P}\left(S \geq x \mid \boldsymbol{\lambda}_{t_0}\right) = e^{-\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\left(1 - e^{-\beta^{\mathsf{c,c}}x}\right) - \frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\left(1 - e^{-\beta^{\mathsf{c,a}}x}\right) - \frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1 - e^{-\beta^{\mathsf{a,c}}x/K_{t_0}}\right) - \frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\left(1 - e^{-\beta^{\mathsf{a,a}}x/K_{t_0}}\right)}, \tag{15}$$

and the probability that no more messages are ever sent in the conversation is

$$\mathrm{P}\left(S = \infty \mid \boldsymbol{\lambda}_{t_0}\right) = e^{-\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} - \frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} - \frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} - \frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}}, \tag{16}$$

where $\boldsymbol{\lambda}_{t_0}$ is as defined in Equation (9).

Although the results in Theorem 1 and Propositions 1, 2, and 3 are stated for the SysBHP model, these are also immediately applicable to the UHP and BHP models since we have defined them in an encapsulatory manner. Similarly, computational procedures developed for the SysBHP will also function for the UHP and BHP. In the next subsection, we will describe two main algorithms supporting this work.

2.4. Overview of Relevant Computational Procedures

To conclude our discussion of our proposed stochastic models of conversational service, let us now give a brief overview of some of the computational methods we use for these processes. We summarize these here for the sake of clarity in our analytical and experimental discussions that follow, but we also describe the estimation algorithm in detail in Appendix B so that others may easily use these models and methods. We use two main types of procedures for the Hawkes-based models: parameter estimation and Monte-Carlo simulation. Our estimation procedure is an adaptation of the log-likelihood-based expectation-maximization (EM) algorithm that has enjoyed much success for Hawkes processes (e.g., Lewis and Mohler 2011, Halpin 2012, Halpin and De Boeck 2013) and was originally developed for general branching processes by Veen and Schoenberg (2008). The procedure is known to be equivalent to projected gradient ascent for the Hawkes process (Lewis and Mohler 2011). Much of its popularity stems from its efficiency, and that is particularly true for our setting thanks to the lack of baseline arrivals. This means that we have a clear identification of what branch or conversation a message is within. For M as the total number of conversations and N_m as the number of messages within conversation m, leveraging the zero-baseline yields a procedure that is $O\left(\sum_{m=1}^{M} N_m^2\right)$ rather than $O\left(\left(\sum_{m=1}^{M} N_m\right)^2\right)$. The EM algorithm is also quite interpretable: what we are doing at each step is calculating the probability that each message is in response to a previous message and then re-expressing the process parameters in terms of these probabilities; this underlying branching structure is the missing data targeted by the EM algorithm in Veen and Schoenberg (2008).

For simulation, here we use a combination of two of the most popular Hawkes process simulation algorithms, the Lewis-Shedler-Ogata thinning-based algorithm (Lewis and Shedler 1979, Ogata 1981) and the Dassios-Zhao exact simulation procedure (Dassios et al. 2013). Each of these has both strengths and weaknesses in addressing the models we have proposed. The former allows us to handle the non-Markovian non-stationarity of the SysBHP model by simulating in each of the different periods of concurrency values. However, the thinning-based methodology is built on identifying an upper-bound for the instantaneous arrival rate of the messages, and this means that it is unable to replicate the end-conditions of models such as ours in which the correspondence rates almost surely converge to 0. By comparison, the immigration-descendant structure of the

Dassios-Zhao procedure is very well structured to handle processes that eventually cease; however, it is not designed to handle non-stationarity. Thus, our hybrid simulation procedures uses both: on all concurrency intervals until the last one of the work shift, we use Lewis-Shedler-Ogata, and then for the final concurrency interval, we employ Dassios-Zhao to complete the replication.

3. Evaluating the Conversational Models on Contact Center Data

To evaluate and compare these stochastic process models of the conversation progression, we will now apply them to true contact center data from industry. We evaluate the accuracy on two main quantities of the conversational stochastic process: the *conversation duration* and the *gap times*. We define the conversation duration as the time elapsed from the start of the conversation with the customer's initial message to the time that the final message is sent. Similarly, we define the gap times as the time elapsed between two consecutive messages. We compare the models to data by simulating 100,000 conversations with the estimated parameters. Following our modeling philosophy that views the customer-agent pairs as homogeneous across conversations, we estimate and evaluate out-of-sample (in which we use 75% of the data for estimation and 25% for evaluation).[4] Before reviewing the parameter estimates, let us first provide summary-level statistics describing our data source.

3.1. Background and Summary Statistics of Contact Center Data

From a communications company's contact center observed during the month of May 2017, we have acquired data containing 337,224 service conversations and a total of 4,964,895 messages. This center operates 24 hours per day, 7 days per week. The average number of new conversations is 602.68 per hour ($SD = 83.59$). The mean number of online agents is 134.69 ($SD = 31.06$), the mean agent concurrency is 4.79 customers per agent ($SD = 2.49$), and the percent of conversations transferred between agents is only 0.09% (291 of the 337,224 conversations). The average conversation duration is 53.48 minutes ($SD = 65.15$); its distribution is given in Figure 4. Figure 5 shows the distribution of the customer gap time and the agent gap time, which have means of 2.58 and 4.26 minutes, respectively ($SD = 9.63, 16.38$, respectively). Each conversation contains an average of 14.72 messages ($SD = 15.02$), out of which 27.9% were written by the customer and 72.1% by the agent. Before normalization to unit mean, each customer message averages 13.14 words ($SD = 16.02$) and each agent message averages 23.0 words ($SD = 22.74$). Hence, after normalization, the word count random variables W^{c} and W^{a} have standard deviations equal to the true distribution's coefficients of variation, $CV^{\mathsf{c}} = 1.22$ and $CV^{\mathsf{a}} = 0.99$. Likewise, before normalization and reverse-mapping, the sentiment scores in the data range from 24 (extremely positive) to -14 (highly negative), with the customer having a relatively neutral mean of 0.102 ($SD = 0.794$) and the agent having a truly neutral mean of 0.00 ($SD = 0.015$).

[4] We have also performed in-sample tests, but the performance was similar and thus we omit it for brevity's sake.

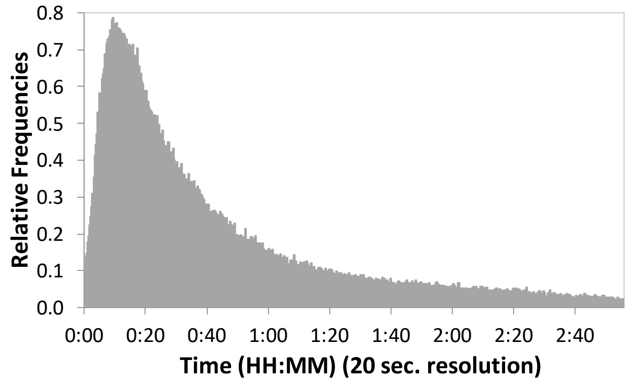
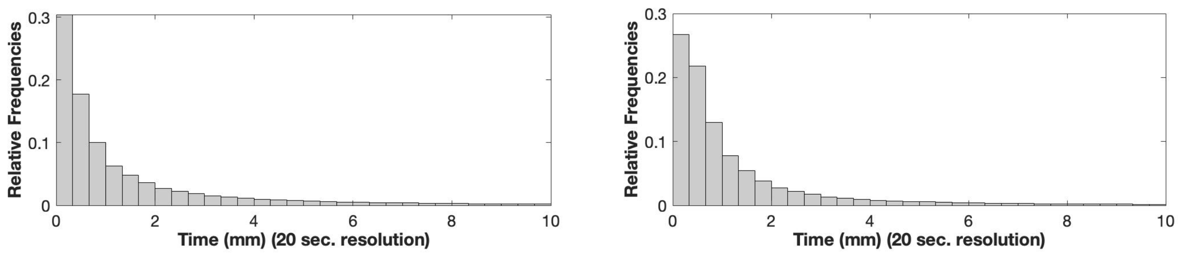


Figure 4 **Conversation duration distribution (All data, May 2017)**



(a) Agent response time distribution (b) Customer response time distribution

Figure 5 **Response time distribution (All data, May 2017)**

3.2. Parameter Estimates and Performance Evaluation of the Stochastic Models

Recall that we want to compare the models proposed in Section 2: the standard and static SES model, the time- or stage-dependent but still stochastically independent SED model, the history- or path-dependent UHP model, the history- and role-dependent BHP model, and the system- and information-driven path- and role-dependent SysBHP model. The first model treats the gaps between messages as independent and identically distributed, and the second views them as independent but non-identical. These two models assume no behavioral effects exist. In contrast, the final three models exhibit conversation path dependence, with each successive Hawkes process model layering levels of behavioral aspects from the two roles in the service co-production, the contents of the messages, and the state of the service operation overall. After fitting each of the co-production models to the data, we give the parameters for all five conversational models, estimated using training data for the out-of-sample test, in Table 1.

With the parameters of each model in hand, we can now also simulate the stochastic processes and compare the resulting distributions with what is seen in the data. To provide a comprehensive perspective on the accuracy of the stochastic models, let us offer a collection of different comparisons. Starting with a visual comparison, in Figure 6, we show plots for the differences between

Table 1 Estimation of Parameters for Each Model from Training Set (May, 1–23, 2017)

Model	Parameters*
SES	$1/\mu = 0.068$. Number of Messages $\sim \mathsf{Geo}(p = 0.065)$
SED	See Table 3 in Appendix C. Number of Messages $\sim \mathsf{NegBin}(r = 1.3,\ p = 0.09)$
UHP	$\alpha = 7.81, \quad \beta = 8.39$
BHP	$\alpha^{\mathsf{c,c}} = 0.89, \alpha^{\mathsf{c,a}} = 14.67, \alpha^{\mathsf{a,c}} = 3.76,\ \alpha^{\mathsf{a,a}} = 20.22,\ \beta^{\mathsf{c,c}} = 3.73, \beta^{\mathsf{c,a}} = 38.35, \beta^{\mathsf{a,c}} = 4.21,\ \beta^{\mathsf{a,a}} = 48.28$
SysBHP	$\alpha_1^{\mathsf{c,c}} = 0.67, \alpha_1^{\mathsf{c,a}} = 14.06, \alpha_1^{\mathsf{a,c}} = 14.79, \alpha_1^{\mathsf{a,a}} = 113.65,\ \beta^{\mathsf{c,c}} = 3.64, \beta^{\mathsf{c,a}} = 38.39, \beta^{\mathsf{a,c}} = 20.33,\ \beta^{\mathsf{a,a}} = 259.72$ $\alpha_2^{\mathsf{c,c}} = 0.18, \alpha_2^{\mathsf{c,a}} = 0.03, \alpha_2^{\mathsf{a,c}} = 2.27, \alpha_2^{\mathsf{a,a}} = 0.07$

*The data and the resulting parameter estimates are on a 1 hour timescale.

the empirical cumulative distribution function (CDF) of the data and the empirical CDF of the simulated stochastic process. Specifically, we consider the differences in CDFs of the conversation duration for all models in Figure 6(a), and likewise for the differences in CDFs for the gap times in Figure 6(b). The closer a curve is to the dashed horizontal line at 0, the closer that model is to the data. These plots are both calculated for the out-of-sample setting, so the durations and gaps considered were not part of the data set from which the process parameters are estimated. Then, Table 2 contains the Kolmogorov-Smirnov (KS) and 1-Wasserstein (W_1) distances between the empirical CDFs of the data and the simulation results. That is, for the empirical CDF of the data $\hat{F}_D$ and of the simulated model $\hat{F}_M$, these distances are

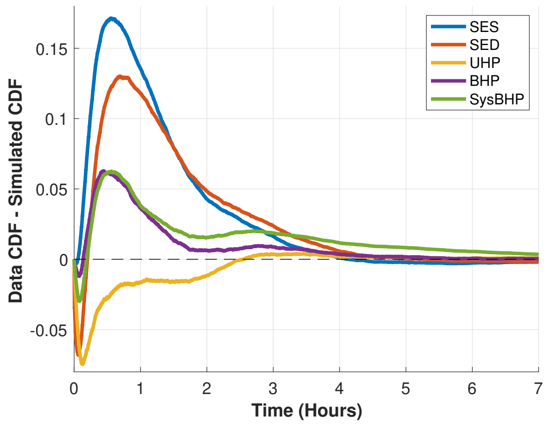
$$\mathrm{KS} = \max_x |\hat{F}_D(x) - \hat{F}_M(x)| \qquad \text{and} \qquad \mathrm{W}_1 = \int_0^\infty |\hat{F}_D(x) - \hat{F}_M(x)| \mathrm{d}x.$$

For relative scales, it can be directly seen that the KS distance is no more than 1, and by the triangle inequality, W_1 is at most the sum of the means of the data and the simulation. Both values are given in Table 2. Naturally, both these distances are intimately related to the curves in Figure 6. For a given model, the KS distance will be the global maximum or minimum of the CDF difference curve, and the W_1 will be the total area between the curve and the dashed line at 0.

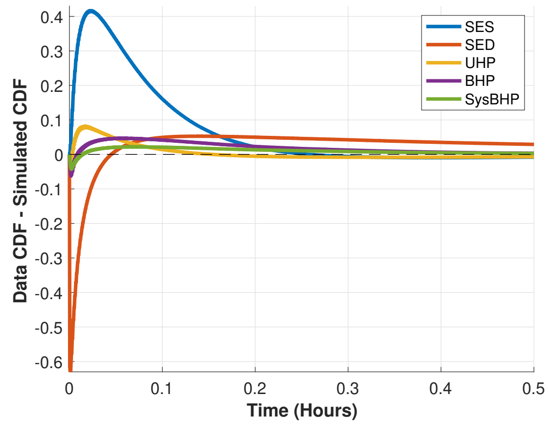
Table 2 Evaluation of Model Fit

	Duration		Gap	
Model	KS	W_1	KS	W_1
SES	0.171	0.254	0.417	0.041
SED	0.130	0.228	0.633	0.046
UHP	0.074	**0.061**	0.096	0.031
BHP	**0.058**	0.078	0.059	0.020
SysBHP	0.063	0.137	**0.043**	**0.006**

In reviewing Figure 6 and Table 2, we see three leaps in performance. First, the plots and distances both show that moving from non-behavioral models, i.e., the static (SES) and time-varying exponential (SED) models, to the history-dependent UHP offers improvement in both the duration fit and the gap-time fit. In the KS distance for the duration, the SES and SED empirical CDFs differ from the data by as much as 0.171 and 0.130, respectively, and worse, for the gap



(a) Conversation Duration



(b) Gap Time

Figure 6 Difference of empirical CDFs for the data and simulated conversation models. Out-of-sample test.

times, these KS distances are 0.417 and 0.633, respectively. In the W_1 distance, the difference between the SES and SED durations and the true durations is even more pronounced, although it is less so for the gaps, which is natural in this metric given the shorter time scale. By comparison, the UHP achieves KS distances of 0.074 for the duration and 0.096 for the gaps, along with an experiment-best 0.061 W_1 distance for the duration. Particularly in comparing the UHP and SED, this improvement is important in that it shows that the conversation's pace and sequencing up to now are valuable service modeling components, and that merely tracking the number of messages so far while having predefined phase-dependent rates does not offer the same level of performance (we will elaborate on the importance of the process history in the next subsection).

We can see a second improvement when moving from the UHP to the role-dependent BHP, particularly so for the gap times. The KS distance decreases to 0.058 for the duration and 0.059 for the gaps; this is the best duration KS score in this evaluation. For W_1 distance, the performance is good albeit not experiment-leading; one can see in Figure 6(a) that the UHP curve aligns to 0 sooner than the BHP does, although the UHP's absolute error is larger. In Figure 6(b), however, the BHP is closer to the data both in extreme and in area. Just as the SES and SED to UHP improvement shows the modeling value of the conversation's history, this improvement when moving to the BHP demonstrates the modeling value of distinguishing the roles within the service co-production. We believe this is a particularly important takeaway for the goals of modeling and analyzing customer and agent behavior, as it shows (a) the different contribution that each role has in the conversation's pace and (b) the influence that the interaction between the two roles can have on the service exchange.

Finally, our third leap comes when moving from the BHP to the SysBHP, revealing itself in the gap time distributions. Although the SysBHP has clear limitations in representing the conversation's duration as a whole from start to finish, it has by far the best performance in the gaps, achieving a KS distance of 0.043 and a 0.006 in W_1 distance. We can confirm this observation visually in Figure 6, where the BHP can be seen to be closer to the data in the duration while the SysBHP is closer in the gap times. This shows us that when using these Hawkes conversational models for decision making over short intervals of time, factoring in the agent's caseload as well as agent and customer behavior can improve model accuracy.

Before proceeding, let us point out a cautionary tale. In reviewing the model fits in terms of the duration, the independent exponential models, SES and SED, seem not so far off of their Hawkes counterparts. However, this can be seen as an empirical example of the law of large numbers limit theorems that self-exciting processes are known to satisfy, see e.g., Bacry et al. (2013), Fierro et al. (2015), Gao and Zhu (2018b), Daw and Pender (2021), and which the independent models should also satisfy. Because the conversational duration is a sum of the gap times, this more closely resembles a law of large numbers limiting object than the individual gap times do themselves. Hence, the mutual closeness of the conversation-duration fits not only should be expected, but also demonstrates the pitfalls of not considering the individual gap times themselves. This adds credence to the granularity of our modeling philosophy of treating the full service duration as a stochastic process rather than a random variable, and our results for the gap-time fits shows that the Hawkes models offer superior representations of reality at this level.

This also speaks to a discrepancy between this manner of model evaluations and our stated modeling goals. We have sought to model a conversation as a dynamic process, so that we can then reason about a conversation's future given its past. Both the duration and gap time comparisons treat the data and the simulations as completely separate. There is, of course, value and insight to this as we have shown in the preceding discussions, but we will also eventually want to test the models' predictive ability operationally. This will be the basis of our data-driven experiment in Section 4. Before doing so, we first want to discuss in more depth the results of the estimated coefficients themselves and show what they can reveal regarding the nature of the co-produced process in reality.

3.3. Interpreting Service Behavior and Relationships from the Process Estimation

Having fit the models to data, let us now inspect these estimated parameters and see what insights they hold for the contact center system. Recalling the definition of the models in Section 2, Theorem 1, and the surrounding discussions, we are interested in the parameters and the responsiveness ratios within each Hawkes conversational model.

History-dependence: In the UHP model, it is known that $\frac{\alpha}{\beta} < 1$ is required for stability, so a responsiveness ratio close to 1 indicates a highly active conversation. This is what we find in the estimated parameters, as $\frac{\alpha}{\beta} = 0.931$. The UHP model can also reveal the level of history-dependence within a conversation in practice. For an approximate analysis, we use Equation (3) and the average gap time ($\Delta = 0.068$ hours). Approximating the i-j message time gap as $A_i - A_j = (i-j)\Delta$, the probability that message i is in response to message $i-1$ is roughly

$$\frac{\alpha}{\alpha + \sum_{k=0}^{i-2} \alpha e^{-\beta\Delta(i-1-k)}} = \frac{1}{\sum_{k=0}^{i-1} e^{-\beta\Delta k}} = \frac{1-e^{-\beta\Delta}}{1-e^{-\beta\Delta i}}.$$

For $\beta = 8.39$ and $\Delta = 0.068$, this probability is less than 50% by the fourth message; it approaches 43.5% as i grows large. While this calculation may be somewhat crude in replacing the gap times by their mean, it is worth recalling that our discussion around Equation (3) suggests that this approximation may be an overestimate of the i to $i-1$ response probability. That is, by evenly spacing the gaps, we are implying that the $(i-1)^{\text{th}}$ message does not occur in a cluster of activity, as there are no clusters. Since the Hawkes process is known to form bursts of activity, we know that these evenly sized spaces are unrealistic and that if message $i-1$ were to occur in a cluster, it would be even less likely that message i is in response to its immediate predecessor. Since the high response ratio estimated for this data implies high activity levels, we expect message clusters to be common.

Role-dependence and the level of co-production: While the UHP is informative for the relevance of history within the service, this single-stream model does not necessarily reveal much more about the service co-production. We can make more nuanced observations on the nature of each role's responsiveness through the BHP and SysBHP models. Starting with the BHP model, we find that the self-production terms (i.e., a party writing a message in response to their own) are relatively low, with $\frac{\alpha^{\mathsf{c},\mathsf{c}}}{\beta^{\mathsf{c},\mathsf{c}}} = 0.239$ for the customer but much higher for the agent: $\frac{\alpha^{\mathsf{a},\mathsf{a}}}{\beta^{\mathsf{a},\mathsf{a}}} = 0.419$. This is as one might expect for the contact center context and resonates with a common practice in call and contact centers that trains employees to keep a constant flow of communication so that the customer will not feel ignored or neglected, being unable to see the agent exert effort to solve her problems. Interestingly though, we find that, although the responsiveness of the agent to the customer is high (as measured by the average number of messages a customer message evokes from the agent: $\frac{\alpha^{\mathsf{a},\mathsf{c}}}{\beta^{\mathsf{a},\mathsf{c}}} = 0.8931$), the customer is not nearly as responsive to the agent in return ($\frac{\alpha^{\mathsf{c},\mathsf{a}}}{\beta^{\mathsf{c},\mathsf{a}}} = 0.383$). It is also interesting to consider this alongside the strong instant impact that the agent messages have on both correspondence rates, with $\alpha^{\mathsf{c},\mathsf{a}} = 14.67$ and $\alpha^{\mathsf{a},\mathsf{a}} = 20.22$. This suggests that, although messages from the agent can significantly alter the current pace of the conversation for the moment, the age-old adage that the customer is the most important remains true. It is the customer messages that drive the long-term communication, even if they may not hold nearly the level of instantaneous effect that the agent messages do.

In the SysBHP model, we can see that this co-production dynamic is again quite pronounced. After modulating the agent's correspondence rate based on her workload and adjusting the jump sizes for the message contents, the values of the parameters change but the responsiveness ratios stay relatively similar to the BHP. This makes sense, as the ratio is independent of time scale. In the style of Theorem 1, we can see that the agent-responding-to-customer co-production response ratio ($\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} = 0.839$) is larger than the two self-production ratios ($\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} = 0.234$ and $\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} = 0.438$) by a good margin, reflecting the collaborative nature of this service and the agent's attentiveness to the customer and to the conversation in general. The customer-responding-to-agent ratio ($\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} = 0.367$) is close to the larger of the two self-production ratios, but it is again clear that it is the customer drives the conversation through eliciting responses from the agent; the agent is hanging on to what the customer writes. Additionally, the agent's propensity to follow up her own comments and maintain a flow of communication is again well represented in this model with the high agent self-production ratio. Because both correspondence rates increase upon the receipt of a message from either party, let us also consider the mean collective response for a given message. That is, the sum $\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} + \frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} = 1.073$ (likewise, $\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} + \frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} = 0.805$) is the mean number of messages that both parties send in response to a given message from the customer (agent). Again, this shows that the customer drives the conversation. On average, the messages she sends create more future activity in the conversation than agent's, even while the customer is the party least likely to respond to herself.

We can also see that the α_1's are much larger than the α_2's, placing more emphasis on the sentiment of a message rather than the word count. This is particularly true for the response streams for agent messages (meaning $\mathsf{c,a}$ and $\mathsf{a,a}$ superscripts), where the α_2's are very small relative to their corresponding α_1's and β's. Practically, for these parameters, a message has to have many more words than average to seriously impact the correspondence rates. It is also worth noting that both sentiment scores are less volatile than the word count distributions, and so this relationship between the α_1's and α_2's shows that the jumps in SysBHP correspondence rates likely do not vary much from agent message to agent message. The word counts of customer messages seem to have a bit stronger of an effect by comparison, as $\alpha_2^{\mathsf{c,c}}$ and $\alpha_2^{\mathsf{a,c}}$ are closer to their α_1 and β counterparts than the agent α_2's were to theirs. It is fitting of the service context that process is more sensitive to the amount of information that the customer writes, considering that the customer is the one who brings the issues to the table in the co-production.

Reciprocity: Another important issue the models capture is the role of reciprocity in the burstiness of the service process. We can measure how each role impacts the other using the process parameters. For example, in the BHP model, recall that when a customer writes a message the agent's message intensity jumps by $\alpha^{\mathsf{a,c}}$ and then diminishes according to $e^{-\beta^{\mathsf{a,c}}s}$ for s time elapsed since the customer's message, i.e., the agent's response rate to this particular customer message s time later is $\alpha^{\mathsf{a,c}}e^{-\beta^{\mathsf{a,c}}s}$. Similarly, in the manner of Proposition 3, the probability that the agent eventually responds to a message sent by the customer just now is $1 - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}}$, while the probability that the agent eventually responds to it given that she has not in the first s time units is $1 - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}e^{-\beta^{\mathsf{a,c}}s}}$. To see the effect of reciprocity in the service co-production, consider the following simple but demonstrative example: one party sends two successive messages and waits for a response, the probability and pace of which are both shaped by the speed (or lack thereof) of those two messages. Suppose that the customer sends two messages to the agent, one now and one some time s later. Assuming the agent does not respond until after both have been sent, the probability that the agent responds to this pair of messages is

$$1 - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1+e^{-\beta^{\mathsf{a,c}}s}\right)}.$$

By comparison, if the customer is slightly slower in her correspondence and takes an additional σ units of time to send the second message, the probability that the agent responds to the pair is instead

$$1 - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1+e^{-\beta^{\mathsf{a,c}}(s+\sigma)}\right)}.$$

Relatively speaking, this delay in the customer's messaging causes a percent decrease in the agent's response probability that can be quantified through

$$\frac{e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1+e^{-\beta^{\mathsf{a,c}}(s+\sigma)}\right)} - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1+e^{-\beta^{\mathsf{a,c}}s}\right)}}{1 - e^{-\frac{\alpha^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1+e^{-\beta^{\mathsf{a,c}}s}\right)}}.$$

This delay can make a pronounced impact on the conversation as a whole. For example, using the estimated parameters of the BHP with s as the observed mean customer response time and σ as its standard deviation, the probability of agent response reduces by 10.63% of the response probability at the mean response time. Likewise, the analogous calculation for increasing the agent's messaging time by 1 SD in a sequence of two messages will reduce the customer response probability by 4.59% of the probability at the mean response time level. Let us also note that this reciprocity is not only consequential for the probability of response, but also for the speed of response. Similar calculations to the above will show that the expected time for one side's response decreases (increases) as the time between the other side's messages decreases (increases) in this pair-of-messages example.

We have now seen that there is value in capturing both the history and the roles of a contact center service conversation. While these behavioral insights into the service co-production are informative, achieving a good fit of the process is only part of our goal. For our models to be of greatest potential value, we also want their representations of customer and agent behavior to be able to inform operational decisions. In the next section, we will apply our models to the managerial problem within a data-driven simulation of the contact center. In doing so, we will show that these same insights around the history and roles of a conversation create the service improvement opportunities without requiring any additional resources.

4. Managerial Implication of Conversational Models: Routing Customers using Predicted Workload

To apply the Hawkes co-production models to decision making in an environment credibly connected to practice, we will now develop a contact center simulation entirely driven from data. In particular, our stochastic conversational models are not at all involved in the randomization. That is, the customer arrival times and the messaging service times, including the times to read and write messages, will all be drawn directly from data. The only role our models will play is in decision making: to which agent should a new conversation be routed and when. We will use the prediction developed in Section 2.3 to differentiate between agents' load.

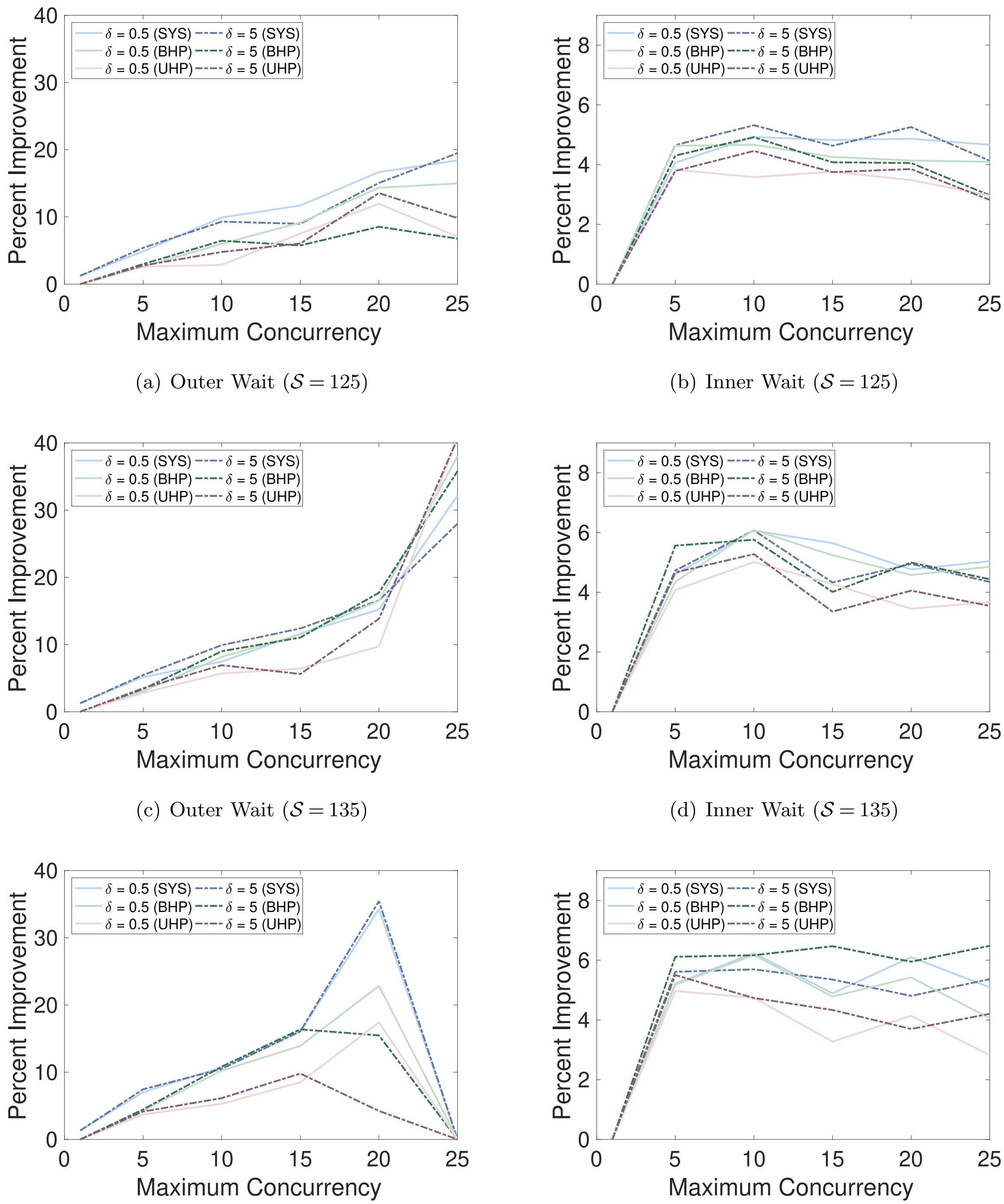
The simulation model we use is a collection of parallel Erlang-R queueing systems with dedicated servers, with similar structure to Campello et al. (2017) (see Figure 1 therein for a visualization of the system with particular distributional assumptions). The Erlang-R model has only mutual-excitation (i.e., exclusively back-and-forth), therefore, we extend it to also incorporate self-excitation (e.g., follow-ups) as follows. Customers arrive and enter an "outer queue" to wait for an assignment to an agent. Once there is an available agent and the customer is assigned, the customer sends her conversation-initializing message and enters the "needy" stage of the system. However, the customer might need to wait in an "inner queue" for the agent to reply if the agent is busy with other customers. Once the agent is available and replies to the customer, the customer enters a "content" stage of the system and stays there for the duration of the customer's response time before returning to the needy state (meaning, waiting in the inner queue). This cycle of customer response time (content state) followed by agent response time (inner wait plus writing time in the needy state) describes mutual-excitation and we will refer to it as a *turn*. Each party can also partake in self-excited behavior by repeating a current stage and writing several messages in succession. The turns alternate back-and-forth between the parties until the conversation completes, according to the total number of turns in the data. Hence, there are two types of waiting recorded: the *outer wait*, in which the customer waits to be paired to an agent, and the *inner wait*, in which the customer waits for a response after sending a message to the agent. In our experiment, the inner wait is reported per each turn and the outer wait is reported per conversation. Like the arrival times, we draw the number of back-and-forth turns of the customer response times directly from data. We also extract from the data a partition of the agent gap times that determines the agent writing times. We preserve the association of the conversation turns, customer response times, and agent writing times in a conversation and sample them collectively, but this does not mean we are sampling the conversation durations exactly as they appear in the data. Because the agent may have a different number or simply a different collection of customers, both the inner and outer waiting times between messages will differ from what originally occurred.

As mentioned, the decisions we consider in this simulation are the routing of new conversations. The routing policies we consider are modified from the "lightest load (LL)" or "join the smallest caseload (JSC)" policy, which has been shown to be asymptotically optimal for some sets of assumptions (see e.g. Luo and Zhang 2013, Tezcan and Zhang 2014, Campello et al. 2017, Long et al. 2019). From industry partners, we understand that this routing algorithm is also very popular in practice. Luo and Zhang (2013) describe the policy as so: "if all agents are serving [the maximum allowable number of] customers, then an arrival has to wait. Otherwise, the arriving customer is assigned to one of the agents who has the 'lightest load' at the time. If there are multiple agents with the same 'lightest load,' one is chosen randomly to serve the arrival." We propose a modest modification of this policy. Instead of breaking ties randomly between multiple agents with the lightest load, we will project the number of upcoming messages in the agents' conversations according to Propositions 1 and 2 and then select the agent with the lowest expectation.

Here, we compare the projection abilities of each of the history dependent models, the UHP, BHP, and SysBHP. This experiment is also conducted out of sample, using this same training and testing sets as used in the model calibration and evaluation in Section 3. We evaluate the projections over three different interval sizes: $\delta = 30$ seconds, $\delta = 5$ minutes, and $\delta = \infty$. We run the simulation and consider the performance in three different staffing settings: $\mathcal{S} = 125$, 135, and 145 servers. The maximum concurrency each agent may have, κ, is varied from 1 to 30 at multiples of 5. It is worth noting that the $\kappa = 1$ and 5 are fairly unrealistic in true contact center contexts (as indicated by Figure 2 and the summary statistics in Section 3.1), but we include them anyway for the sake of comparison.

The comparison between the routing performance delivered by each model (and by the unmodified LL algorithm) is presented in Figure 7 and Tables 4–8 (see Appendix). Specifically, Figure 7 contains the percent improvement in mean inner and outer wait for each model-boosted routing policy, where percent improvement is calculated in relation to the standard implementation of the lightest load policy.[5] As a convention, percent improvement when both values are 0 is set to be

[5] Due to the memoryless property of the exponential and geometric distributions, random tie-breaking in lightest load is equivalent to projecting future activity according to the SES model. Hence, these improvements can also be thought of as relative to SES.



(a) Outer Wait ($\mathcal{S} = 125$)

(b) Inner Wait ($\mathcal{S} = 125$)

(c) Outer Wait ($\mathcal{S} = 135$)

(d) Inner Wait ($\mathcal{S} = 135$)

(e) Outer Wait ($\mathcal{S} = 145$)

(f) Inner Wait ($\mathcal{S} = 145$)

Figure 7 **Percent improvement of wait time metrics for the lightest load policy with BHP and SysBHP projections**

0%. To not crowd the visualizations, Figure 7 excludes the infinite horizon projection ($\delta = \infty$) and the largest concurrency allowance ($\kappa = 30$), but these values can all be found in the tables. Table 4 (Table 5) contains all the values and percent improvements of the mean inner (outer) waits, whereas Table 6 (Table 7) contains the same for the standard deviations of the inner (outer) waits. Finally, Table 8 contains the probability of outer wait for each experiment setting.

As can be seen throughout these plots and tables, each of the Hawkes models offer substantial improvement to the lightest load. While all do well, we can see that most often the SysBHP model offers the strongest performance, demonstrating the operational benefit of addressing the customer and agent relationships and the behavioral features of their message contexts. Specifically, the SysBHP can improve the inner and outer wait times by over 6% and 20%, respectively. In fact, Table 5 shows that the SysBHP can reduce the outer wait times by an even larger amount, but Table 8 shows that these particularly extreme reductions often occur in scenarios when very few customers wait before service; hence, we focus on the cases in which wait is more common. Across the staffing-concurrency-interval scenarios we consider, the inner wait decreases by an average of 4.1% and the outer wait decreases by an average of 11.5% when using the policy with SysBHP projections. Supporting our assertion that the customer and agent relationships are vital components of service progression models, the BHP policies also perform quite well across the board; in some cases, it can be seen to rival or surpass the SysBHP. Averaging across all the scenarios, the BHP model decreases the mean inner waits by 4.0% and outer waits by 8.5%. The UHP lags behind the other two but still offers valuable improvements, reducing the lightest load scenario averages by 3.4% and 7.2% in mean inner and out wait, respectively. This further showcases the operational benefits of leveraging the conversation's dependence on its own history.

Let us note that these policies are not just decreasing the average waiting time, but reducing system variability as well. Tables 6 and 7 show that the Hawkes projection policies can also reduce the standard deviation of the waits, often by 3-6% of the standard deviations under the unadjusted lightest load policy. Furthermore, all differences in mean between the standard lightest load routing and the model projection policies are statistically significant (except for when there is no wait, such as inner wait at $\kappa = 1$ or outer wait at $\mathcal{S} = 145$ and $\kappa = 25$). Additionally, for each model, we see no sizable difference between using the short or long time window for the load estimation (i.e., different δ's), although the SysBHP's superiority may be most pronounced at $\delta = 0.5$ minutes, aligning with its relative strength in representing the gap time distributions. The similarity between projections within each model is understandable as all measures use the same information under the same assumption of no concurrency change in the near future. In addition, the algorithm only compares between agents, and if one agent is more loaded than another at time $t_0 + \epsilon$, he will continue to be so until a change in the system.

Before closing, let us remark that the policies we have shown are just a humble first step. There may very well be policies designed for the Hawkes models that can surpass the performance of the modified policies we have just presented. We leave such further developments, which are of particular interest to us, to future research. For now, we offer these projections on top of the lightest load policy as a way of boosting system performance at no added cost. That is, by employing our Hawkes conversational models, one can improve contact center operations without allocating any additional resources.

5. Discussion and Conclusion

In this paper, we have studied the co-production of service in contact centers by developing conversational models based on Hawkes processes. At their most complex, these models capture customer-agent relationships while also incorporating operational and behavioral features such as the agent's concurrency, the number of words in a message, and the message's sentiment. Our models give us both a representation for the service process and a way to predict the process's future activity, which can be used operationally for strategic decisions such as routing new conversations. As we observe in our case study on an industry data set, these models fit data with high accuracy. Through calibrating the models to industry data, we have also found insights on the nature of the service co-production. For instance, through interpretation of the estimated parameters, we have showcased the customer's central role in the service co-production, as her queries drive the conversation.

Operationally, our most important takeaway is that the Hawkes conversational models can be used to improve managerial decision making. We demonstrated this in our data-driven routing simulation, in which we compared policies known in the literature to adaptations that use the insights of these co-production models. While the modifications were modest in concept, they had outstanding impact, dropping wait times by over 20%. On a general level, our experiments show that bringing operational and behavioral features into stochastic process service models can improve system performance. We believe this holds high practical value, as implementing the modified policy is free to the contact center: no additional resources or staff are required for improvement. It is also worth noting that these predictive models do not use any features of the data other than the message timestamps, the agent's number of assigned conversations, the number of words written per message, and the sentiment expressed in each message. Thus, they are both light in implementation and respectful of the privacy of the conversation, since we do not need to know exactly what was said by either party.

We have been open throughout this paper that while our models offer refined service representations and improved operational performance, there are many opportunities for advancement. We view our contributions as important first steps for this direction of service interaction stochastic models. As an interesting future direction of modeling, we can note that in highly asynchronous conversations like email, Hawkes processes with non-exponential decay structures have been used to some modeling success. Thus, it would be interesting to compare the performance of such models in this moderately asynchronous setting. Non-exponential kernels will require more computationally intensive estimation techniques, but could also lead to curves nearer to 0 in Figure 6. We are also interested in adding topic analysis features to combine the conversational approach we took with the activity-based approach considered previously in Mandelbaum and Reiman (1998). This adaptation will require a significant generalization to incorporate categorical variables into the model. Since we have noted that our model does not "read" the conversations, it would be interesting to compare our strong performance with predictions of activity using natural language processing.

Furthermore, we have assumed independence among the agent's conversations, or more specifically, conditional independence given the agent's concurrency. This aligns with our modeling focus of representing the service conversation, rather than the service system, as a stochastic process, but this scope could be widened. One could attempt to relax this limitation through a multivariate Hawkes process model that addresses the possible dependence these conversations may have through their shared agent; this also would make an interesting direction of future work. On the operational side, we have shown that our models can improve the popular lightest load routing policy through projecting a conversation's future activity, but we believe there is much more potential here. As we have noted, there may be a different approach designed for the Hawkes models first rather than for the queueing models first. That is, one could imagine "lightest load" policies that view the agent's workload through the lens of the Hawkes conversational models rather than through traditional queueing notions like the number of customers. For instance, rather than using the maximum concurrency threshold like in lightest load, there could instead be a maximum number of expected messages. In general, we are interested in investigating routing policies in the Hawkes process space and analyzing their performance. Of course, other managerial problems like contact center staffing may also be amenable to this perspective, and we look forward to seeing what other insights the Hawkes conversational models can offer.

Acknowledgements

This research is supported by the Israel Science Foundation (ISF) through grant 336/19 and by the National Science Foundation (NSF) under grant DGE-1650441.

References

Abergel F, Jedidi A (2015) Long-time behavior of a Hawkes process–based limit order book. *SIAM Journal on Financial Mathematics* 6(1):1026–1043.

Altman D, Yom-Tov GB, Olivares M, Ashtar S, Rafaeli A (2020) Do customer emotions affect agent speed? An empirical study of emotional load in online customer contact centers, Forthcoming in Manufacturing & Service Operations Management.

Ashtar S, Rafaeli A, Yom-Tov GB, Akiva N (2021) Reciprocal influences between customers and employees in emotions expressed during interpersonal service interactions, working paper.

Bacry E, Delattre S, Hoffmann M, Muzy JF (2013) Some limit theorems for Hawkes processes and application to financial statistics. *Stochastic Processes and their Applications* 123(7):2475–2499.

Bacry E, Mastromatteo I, Muzy JF (2015) Hawkes processes in finance. *Market Microstructure and Liquidity* 1(01):1550005.

Batt RJ, Terwiesch C (2015) Waiting patiently: An empirical study of queue abandonment in an emergency department. *Management Science* 61(1):39–59.

Borst S, Boxma O, Hegde N (2005) Sojourn times in finite-capacity Processor-Sharing queues. *NGI 2005 - Next Generation Internet Networks: Traffic Engineering.*

Bray RL, Coviello D, Ichino A, Persico N (2016) Multitasking, multiarmed bandits, and the Italian judiciary. *Manufacturing & Service Operations Management* 18(4):545–558.

Campello F, Ingolfsson A, Shumsky RA (2017) Queueing models of case managers. *Management Science* 63(3):882–900.

Castellanos A, Yom-Tov GB, Goldberg Y (2019) Silent abandonment in contact centers: Estimating customer patience with uncertain data, working paper.

Chen X (2020) Perfect sampling of Hawkes processes and queues with Hawkes arrivals. *arXiv preprint arXiv:2002.06369* .

Dassios A, Zhao H, et al. (2013) Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability* 18.

Davis MH (1984) Piecewise-deterministic Markov processes: a general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)* 46(3):353–376.

Daw A, Pender J (2018) Queues driven by Hawkes processes. *Stochastic Systems* 8(3):192–229.

Daw A, Pender J (2021) An ephemerally self-exciting point process. *arXiv preprint arXiv:1811.04282* .

Delasay M, Ingolfsson A, Kolfal B, Schultz K (2019) Load effect on service times. *European Journal of Operational Research* 279(3):673–686.

Dong J, Feldman P, Yom-Tov GB (2015) Service system with slowdowns: Potential failures and proposed solutions. *Operations Research* 63(2):305–324.

Embrechts P, Liniger T, Lin L (2011) Multivariate Hawkes processes: An application to financial data. *Journal of Applied Probability* 48(A):367–368.

Fierro R, Leiva V, Møller J (2015) The Hawkes process with different exciting functions and its asymptotic behavior. *Journal of Applied Probability* 52(1):37–54.

Fox EW, Short MB, Schoenberg FP, Coronges KD, Bertozzi AL (2016) Modeling e-mail networks and inferring leadership using self-exciting point processes. *Journal of the American Statistical Association* 111(514):564–584.

Freedman JL, Fraser SC (1966) Compliance without pressure: The foot-in-the-door technique. *Journal of Personality and Social Psychology* .

Fuchs VR (1968) The service economy. *NBER Books* .

Gans N, Liu N, Mandelbaum A, Shen H, Ye H (2010) Service times in call centers: Agent heterogeneity and learning with some operational consequences. *IMS Collections. Borrowing Strength: Theory Powering Applications – A Festschrift for Lawrence D. Brown*, volume 6, 99—123 (Institute of Mathematical Statistics).

Gao X, Zhu L (2018a) Functional central limit theorems for stationary Hawkes processes and application to infinite-server queues. *Queueing Systems* 90(1-2):161–206.

Gao X, Zhu L (2018b) Limit theorems for Markovian Hawkes processes with a large initial intensity. *Stochastic Processes and their Applications* 128(11):3807–3839.

Goes PB, Ilk N, Lin M, Zhao JL (2018) When more is less: Field evidence on unintended consequences of multitasking. *Management Science* 64(7):2973–3468.

Guo X, Hu A, Xu R, Zhang J (2018) Consistency and computation of regularized MLEs for multivariate Hawkes processes. *arXiv preprint arXiv:1810.02955* .

Halpin PF (2012) An EM algorithm for Hawkes process. *Psychometrika* 2.

Halpin PF, De Boeck P (2013) Modelling dyadic interaction with Hawkes processes. *Psychometrika* 78(4):793–814.

Hawkes AG (1971) Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58(1):83–90.

Hawkes AG, Oakes D (1974) A cluster process representation of a self-exciting process. *Journal of Applied Probability* 11(3):493–503.

Ilk N (2020) The impact of waiting on customer responsiveness: Field evidence from a live-chat contact center, working paper.

Jennings OB, Pender J (2016) Comparisons of ticket and standard queues. *Queueing Systems* 84(1-2):145–202.

Kc DS (2013) Does multitasking improve performance? Evidence from the emergency department. *Manufacturing & Service Operations Management* 16(2):168–183.

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

Kirchner M, Bercher A (2018) A nonparametric estimation procedure for the Hawkes process: comparison with maximum likelihood estimation. *Journal of Statistical Computation and Simulation* 88(6):1106–1116.

Koops D, Saxena M, Boxma O, Mandjes M (2018) Infinite-server queues with Hawkes input. *Journal of Applied Probability* 55(3):920–943.

Laub PJ, Taimre T, Pollett PK (2015) Hawkes processes. *arXiv preprint arXiv:1507.02822* .

Lewis E, Mohler G (2011) A nonparametric EM algorithm for multiscale Hawkes processes. *Journal of Nonparametric Statistics* 1(1):1–20.

Lewis PW, Shedler GS (1979) Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* 26(3):403–413.

Long Z, Tezcan T, Zhang J (2019) Customer service chat systems with general service and patience times, working paper.

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

Mandelbaum A, Reiman MI (1998) On pooling in queueing networks. *Management Science* 44(7):971–981.

Massoulié L (1998) Stability results for a general class of interacting point processes dynamics, and applications. *Stochastic Processes and Their Applications* 75(1):1–30.

Ogata Y (1981) On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory* 27(1):23–31.

Ogata Y (1988) Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association* 83(401):9–27.

Ozaki T (1979) Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics* 31(1):145–155.

Rafaeli A, Yom-Tov G, Ashtar S, Altman D (2020) Opportunities, tools and new insights: Evidence on emotions in service from analyses of digital traces data. Härtel CEJ, Zerbe WJ, Ashkanasy NM, eds., *Emotions and Service in the Digital Age (Research on Emotions in Organizations, Vol 16)*, 105–136 (Emerald Publishing Limited, UK).

RingCentral (2012) Texting for work on the rise per RingCentral survey. Press Release, URL https://web.archive.org/web/20150919105852/https://www.ringcentral.com/whyringcentral/company/pressreleases/pressreleases-2012/131212.html.

Rizoiu MA, Lee Y, Mishra S, Xie L (2017) Hawkes processes for events in social media. *Frontiers of Multimedia Research*, 191–218 (Association for Computing Machinery and Morgan & Claypool).

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L (2018) SIR-Hawkes: Linking epidemic models and Hawkes processes to model diffusions in finite populations. *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 419–428 (International World Wide Web Conferences Steering Committee).

Roels G (2014) Optimal design of coproductive services: Interaction and work allocation. *Manufacturing and Service Operations Management* 16(4):578–594.

Salehi F, Trouleau W, Grossglauser M, Thiran P (2019) Learning Hawkes processes from a handful of events. *Advances in Neural Information Processing Systems*, 12694–12704.

Tan XJ, Wang Y, Tan Y (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, Zhang J (2014) Routing and staffing in customer service chat systems with impatient customers. *Operations Research* 62(4):943–956.

van Leeuwaarden JS, Mathijsen BW, Sloothaak F, Yom-Tov GB (2017) The restricted Erlang-R queue: Finite-size effects in service systems with returning customers, working paper.

Veen A, Schoenberg FP (2008) Estimation of space–time branching process models in seismology using an EM–type algorithm. *Journal of the American Statistical Association* 103(482):614–624.

Wu A, Bassamboo A, Perry O (2019) Service systems with dependent service and patience times. *Management Science* 65(3):1151–1172.

Xu K, Chan CW (2016) Using future information to reduce waiting times in the emergency department via diversion. *Manufacturing & Service Operations Management* 18(3):314–331.

Xu SH, Gao L, Ou J (2007) Service performance analysis and improvement for a ticket queue with balking customers. *Management Science* 53(6):971–990.

Yom-Tov GB, Ashtar S, Altman D, Natapov M, Barkay N, Westphal M, Rafaeli A (2018) Customer sentiment in web-based service interactions: Automated analyses and new insights. *In WWW '18 Companion: The 2018 Web Conference Companion, April 23–27*, 8 pages (New York, NY, USA: ACM).

Yom-Tov GB, Mandelbaum A (2014) Erlang-R: A time-varying queue with reentrant customers, in support of healthcare staffing. *Manufacturing & Service Operations Management* 16(2):283–299.

Yom-Tov GB, Yedidsion L, Xie Y (2020) An invitation control policy for proactive service systems: Balancing efficiency, value and service level. *Manufacturing & Service Operations Management* .

Appendix

A. Proofs and Technical Lemmas

There are four different decay rates, so to analyze the Hawkes models with Markov process tools we need to treat the correspondence rates as four subprocesses: $\lambda_t^{\mathsf{c},\mathsf{c}}$, $\lambda_t^{\mathsf{c},\mathsf{a}}$, $\lambda_t^{\mathsf{a},\mathsf{c}}$, and $\lambda_t^{\mathsf{a},\mathsf{a}}$. These also follow the "in-from" convention for the superscripts like we defined for α and β, e.g., $\lambda_t^{\mathsf{c},\mathsf{a}}$ is the rate of customer messages responding to messages from the agent. In the BHP model, we can quickly observe from Equations (4) and (5) that for $i, j \in \{\mathsf{c}, \mathsf{a}\}$, $\lambda_t^{i,j}$ will jump up by amount $\alpha^{i,j}$ upon any message sent by party j and otherwise will decay at rate $\beta^{i,j}$. In the SysBHP model, these behavioral dynamics are modified by the functions of the system information in Equations (6) and (7). That is, in the customer and agent correspondence rates alike, the sizes of the response-rate jumps are multiplied by the function of the arriving message's word count and sentiment. Then, in the agent's correspondence rate subprocesses, both the jump sizes and the decay rates are modified by the function of the current concurrency level. Because the word and sentiment sequences are independent from the correspondence processes and because the concurrency is known, this implies that the infinitesimal generator can be expressed as follows in Lemma 1.

LEMMA 1 **(Dynkin's Formula)**. *Suppose that* $t \in [t_0, t_1)$ *for* $t_0 < t_1$ *in which* K_t *is unchanged throughout the interval and* $\lambda_{t_0}^{\mathsf{c},\mathsf{c}}$, $\lambda_{t_0}^{\mathsf{c},\mathsf{a}}$, $\lambda_{t_0}^{\mathsf{a},\mathsf{c}}$, *and* $\lambda_{t_0}^{\mathsf{a},\mathsf{a}}$ *are known. Let* $\boldsymbol{\lambda}_t = [\lambda_t^{\mathsf{c},\mathsf{c}}, \lambda_t^{\mathsf{c},\mathsf{a}}, \lambda_t^{\mathsf{a},\mathsf{c}}, \lambda_t^{\mathsf{a},\mathsf{a}}]^\mathsf{T}$ *and* $\mathbf{N}_t = [N_t^{\mathsf{c},\mathsf{c}}, N_t^{\mathsf{c},\mathsf{a}}, N_t^{\mathsf{a},\mathsf{c}}, N_t^{\mathsf{a},\mathsf{a}}]^\mathsf{T}$ *Then, for a sufficiently regular function* $f : \mathbb{R}_+^4 \times \mathbb{N}^4 \to \mathbb{R}$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right] = \mathrm{E}\left[\mathcal{L}f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right]$$

where

$$\begin{aligned}
\mathcal{L}f(\boldsymbol{\lambda}_t, \mathbf{N}_t) = {} & \lambda_t^{\mathsf{c},\mathsf{c}}\mathrm{E}_{S^\mathsf{c},W^\mathsf{c}}\left[f\left(\boldsymbol{\lambda}_t + [\alpha_1^{\mathsf{c},\mathsf{c}}S^\mathsf{c} + \alpha_2^{\mathsf{c},\mathsf{c}}W^\mathsf{c},\, 0,\, \alpha_1^{\mathsf{a},\mathsf{c}}S^\mathsf{c} + \alpha_2^{\mathsf{a},\mathsf{c}}W^\mathsf{c},\, 0]^\mathsf{T}, \mathbf{N}_t + [1,\, 0,\, 0,\, 0]^\mathsf{T}\right) - f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right] \\
& + \lambda_t^{\mathsf{c},\mathsf{a}}\mathrm{E}_{S^\mathsf{c},W^\mathsf{c}}\left[f\left(\boldsymbol{\lambda}_t + [\alpha_1^{\mathsf{c},\mathsf{c}}S^\mathsf{c} + \alpha_2^{\mathsf{c},\mathsf{c}}W^\mathsf{c},\, 0,\, \alpha_1^{\mathsf{a},\mathsf{c}}S^\mathsf{c} + \alpha_2^{\mathsf{a},\mathsf{c}}W^\mathsf{c},\, 0]^\mathsf{T}, \mathbf{N}_t + [0,\, 1,\, 0,\, 0]^\mathsf{T}\right) - f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right] \\
& + \lambda_t^{\mathsf{a},\mathsf{c}}\mathrm{E}_{S^\mathsf{a},W^\mathsf{a}}\left[f\left(\boldsymbol{\lambda}_t + \left[0, \frac{\alpha_1^{\mathsf{c},\mathsf{a}}S^\mathsf{a} + \alpha_2^{\mathsf{c},\mathsf{a}}W^\mathsf{a}}{K_t},\, 0,\, \frac{\alpha_1^{\mathsf{a},\mathsf{a}}S^\mathsf{a} + \alpha_2^{\mathsf{a},\mathsf{a}}W^\mathsf{a}}{K_t}\right]^\mathsf{T}, \mathbf{N}_t + [0,\, 0,\, 1,\, 0]^\mathsf{T}\right) - f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right] \\
& + \lambda_t^{\mathsf{c},\mathsf{a}}\mathrm{E}_{S^\mathsf{c},W^\mathsf{c}}\left[f\left(\boldsymbol{\lambda}_t + \left[0, \frac{\alpha_1^{\mathsf{c},\mathsf{a}}S^\mathsf{a} + \alpha_2^{\mathsf{c},\mathsf{a}}W^\mathsf{a}}{K_t},\, 0,\, \frac{\alpha_1^{\mathsf{a},\mathsf{a}}S^\mathsf{a} + \alpha_2^{\mathsf{a},\mathsf{a}}W^\mathsf{a}}{K_t}\right]^\mathsf{T}, \mathbf{N}_t + [0,\, 0,\, 0,\, 1]^\mathsf{T}\right) - f(\boldsymbol{\lambda}_t, \mathbf{N}_t)\right] \\
& - \beta^{\mathsf{c},\mathsf{c}}\lambda_t^{\mathsf{c},\mathsf{c}}f_{\partial 1}(\boldsymbol{\lambda}_t, \mathbf{N}_t) - \beta^{\mathsf{c},\mathsf{a}}\lambda_t^{\mathsf{c},\mathsf{a}}f_{\partial 2}(\boldsymbol{\lambda}_t, \mathbf{N}_t) - \frac{\beta^{\mathsf{a},\mathsf{c}}}{K_t}\lambda_t^{\mathsf{a},\mathsf{c}}f_{\partial 3}(\boldsymbol{\lambda}_t, \mathbf{N}_t) - \frac{\beta^{\mathsf{a},\mathsf{a}}}{K_t}\lambda_t^{\mathsf{a},\mathsf{a}}f_{\partial 4}(\boldsymbol{\lambda}_t, \mathbf{N}_t),
\end{aligned} \tag{17}$$

with $f_{\delta i}(\cdot)$ *as the partial derivative of* $f(\cdot)$ *with respect to the* i^{th} *coordinate and where* $\mathrm{E}_{X,Y}[\cdot]$ *is the expectation taken relative to the random variables* X *and* Y.

Proof. See, e.g., Section 5 of Davis (1984), and Theorem 5.5 therein for the full regularity conditions on the function $f(\cdot)$. □

As we have discussed, the SysBHP is not a Markov process in general. This is because updates to the concurrency require the conversation's full history in order to update the agent correspondence rate. However, in an interval in which the concurrency is unchanged, the SysBHP is a Markov process, as in these ranges, its behavior is equivalent to a marked BHP model. It is in such intervals that will we apply Markov process tools to this model. As for the regularity conditions on the function $f(\cdot)$, in this paper we are only concerned with the mean. Hence, $f(\cdot)$ is the identity function for one of the subprocesses, and such a function is compactly supported.

A.1. Proof of Theorem 1

Proof. From Theorem 5 of Massoulié (1998) (or Proposition 2.3 of Abergel and Jedidi (2015) for the former applied to a relevant specific case), the BHP model will be stable if the spectral radius of the ratio matrix

$$\mathbf{R} = \begin{bmatrix} \frac{\alpha^{\mathsf{c},\mathsf{c}}}{\beta^{\mathsf{c},\mathsf{c}}} & \frac{\alpha^{\mathsf{c},\mathsf{a}}}{\beta^{\mathsf{c},\mathsf{a}}} \\ \frac{\alpha^{\mathsf{a},\mathsf{c}}}{\beta^{\mathsf{a},\mathsf{c}}} & \frac{\alpha^{\mathsf{a},\mathsf{a}}}{\beta^{\mathsf{a},\mathsf{a}}} \end{bmatrix}$$

is strictly less than 1, meaning that both eigenvalues of $\mathbf{R}$ are less than 1 in absolute value. We can note that the eigenvalues of $\mathbf{R}$ will both be strictly less than 1 if and only if the eigenvalues of $\mathbf{R} - \mathbf{I}$ are both less than 0, and likewise, the eigenvalues of $\mathbf{R}$ will both be strictly greater than -1 if and only if the eigenvalues of $\mathbf{R}+\mathbf{I}$ are both strictly positive. Now, $\mathsf{det}(\mathbf{R}-\mathbf{I}) = (1-\frac{\alpha^{\mathsf{c},\mathsf{c}}}{\beta^{\mathsf{c},\mathsf{c}}})(1-\frac{\alpha^{\mathsf{a},\mathsf{a}}}{\beta^{\mathsf{a},\mathsf{a}}}) - \frac{\alpha^{\mathsf{c},\mathsf{a}}}{\beta^{\mathsf{c},\mathsf{a}}}\frac{\alpha^{\mathsf{a},\mathsf{c}}}{\beta^{\mathsf{a},\mathsf{c}}}$ and $\mathsf{trace}(\mathbf{R}-\mathbf{I}) = \frac{\alpha^{\mathsf{c},\mathsf{c}}}{\beta^{\mathsf{c},\mathsf{c}}} + \frac{\alpha^{\mathsf{a},\mathsf{a}}}{\beta^{\mathsf{a},\mathsf{a}}} - 2$. If

$$\frac{\alpha^{\mathsf{c},\mathsf{a}}}{\beta^{\mathsf{c},\mathsf{a}}}\frac{\alpha^{\mathsf{a},\mathsf{c}}}{\beta^{\mathsf{a},\mathsf{c}}} < \left(1-\frac{\alpha^{\mathsf{c},\mathsf{c}}}{\beta^{\mathsf{c},\mathsf{c}}}\right)\left(1-\frac{\alpha^{\mathsf{a},\mathsf{a}}}{\beta^{\mathsf{a},\mathsf{a}}}\right), \tag{18}$$

then the determinant of $\mathbf{R}-\mathbf{I}$ will of course be positive, but (18) also implies that the trace will be negative since every α and β must be positive. This similarly implies that both the determinant and the trace of $\mathbf{R}+\mathbf{I}$ are positive, and hence (18) implies that the BHP is stable.

To now extend this to the SysBHP, let us construct an alternate BHP model (abbreviated ABHP) in which the jump size in the customer correspondence rate upon receipt of a message from party $j \in \{\mathsf{c},\mathsf{a}\}$ is $\bar{\alpha}^{\mathsf{c},j}$, as opposed to simply $\alpha^{\mathsf{c},j}$ for the standard BHP or $\alpha_1^{\mathsf{c},j}S^j + \alpha_2^{\mathsf{c},j}W^j$ in the true SysBHP. Likewise, let the ABHP be defined so that the jump size in the ABHP agent correspondence rate is $\bar{\alpha}^{\mathsf{a},j}/K_t$. Let $\tilde{\lambda}_t^{\mathsf{c}}$ and $\tilde{\lambda}_t^{\mathsf{a}}$ respectively be the customer and agent correspondence rates for this alternate model, and assume that the ABHP decay rates are the same as in the SysBHP model, i.e. $\beta^{\mathsf{c},\mathsf{c}}$, $\beta^{\mathsf{c},\mathsf{a}}$, $\beta^{\mathsf{a},\mathsf{c}}/K_t$, and $\beta^{\mathsf{a},\mathsf{a}}/K_t$. Then, the proceeding arguments yield that (8) implies that the ABHP is stable. Now, let us suppose further that the SysBHP and ABHP are defined with the same initial values, i.e., $\tilde{\lambda}_0^{\mathsf{c},\mathsf{c}} = \lambda_0^{\mathsf{c},\mathsf{c}}$, $\tilde{\lambda}_0^{\mathsf{c},\mathsf{a}} = \lambda_0^{\mathsf{c},\mathsf{a}}$, $\tilde{\lambda}_0^{\mathsf{a},\mathsf{c}} = \lambda_0^{\mathsf{a},\mathsf{c}}$, and $\tilde{\lambda}_0^{\mathsf{a},\mathsf{a}} = \lambda_0^{\mathsf{a},\mathsf{a}}$. The processes

$$\lambda_t^{\mathsf{c}} - \tilde{\lambda}_t^{\mathsf{c}} \qquad \text{and} \qquad \lambda_t^{\mathsf{a}} - \tilde{\lambda}_t^{\mathsf{a}}$$

are zero-mean martingales, as the corresponding decay rates are equal and the jump sizes of the ABHP are equal to the mean jump sizes of the SysBHP. Therefore, Doob's martingale convergence theorem yields that since the ABHP is stable given (8), the SysBHP is as well. Note that the preceding arguments apply regardless of the value of K_t, and this is reflected in the fact (8) does not depend on K_t.

Finally, to show the almost sure finiteness of the conversation, let us first identify the limiting mean of the ABHP. From Equation (21) of Hawkes' original work (Hawkes 1971), a stationary bivariate Hawkes process with ratio matrix $\mathbf{R}$ (as defined at the beginning of the proof) and baseline intensity vector $\boldsymbol{\nu} \in \mathbb{R}_+^2$ will have steady-state mean intensity equal to $(\mathbf{I}-\mathbf{R})^{-1}\boldsymbol{\nu}$. Since the conversational models we have proposed have no baseline intensity, we have $\boldsymbol{\nu} = 0$. Thus, $\lim_{t\to\infty} \mathrm{E}\left[\tilde{\lambda}_t^{\mathsf{c}}\right] = \lim_{t\to\infty} \mathrm{E}\left[\tilde{\lambda}_t^{\mathsf{a}}\right] = 0$. By construction, $\mathrm{E}\left[\lambda_t^{\mathsf{c}}\right] = \mathrm{E}\left[\tilde{\lambda}_t^{\mathsf{c}}\right]$ and $\mathrm{E}\left[\lambda_t^{\mathsf{a}}\right] = \mathrm{E}\left[\tilde{\lambda}_t^{\mathsf{a}}\right]$, so we also have that $\lim_{t\to\infty} \mathrm{E}\left[\lambda_t^{\mathsf{c}}\right] = \lim_{t\to\infty} \mathrm{E}\left[\lambda_t^{\mathsf{a}}\right] = 0$. Now, because λ_t^{c} and λ_t^{a} are non-negative almost surely, the zero mean implies that $\lim_{t\to\infty} \lambda_t^{\mathsf{c}} = 0$ and $\lim_{t\to\infty} \lambda_t^{\mathsf{a}} = 0$ almost surely as well. Since these correspondence rates are the intensities of the counting process for the total number of messages N_t, this further implies that with probability 1 there will be only finitely many messages, thus completing the proof. □

A.2. Proof of Proposition 1

Proof. Because $\mathrm{E}\left[S_1^{\mathsf{c}}\right] = \mathrm{E}\left[W_1^{\mathsf{c}}\right] = \mathrm{E}\left[S_1^{\mathsf{a}}\right] = \mathrm{E}\left[W_1^{\mathsf{a}}\right] = 1$, Lemma 1 yields that the means of the SysBHP subprocesses satisfy the following system of ordinary differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\lambda_t^{\mathsf{c},\mathsf{c}}\right] = \bar{\alpha}^{\mathsf{c},\mathsf{c}}\left(\mathrm{E}\left[\lambda_t^{\mathsf{c},\mathsf{c}}\right] + \mathrm{E}\left[\lambda_t^{\mathsf{c},\mathsf{a}}\right]\right) - \beta^{\mathsf{c},\mathsf{c}}\mathrm{E}\left[\lambda_t^{\mathsf{c},\mathsf{c}}\right],$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\lambda_t^{\mathsf{c,a}}\right] = \bar{\alpha}^{\mathsf{c,a}}\left(\mathrm{E}\left[\lambda_t^{\mathsf{a,c}}\right] + \mathrm{E}\left[\lambda_t^{\mathsf{a,a}}\right]\right) - \beta^{\mathsf{c,a}}\mathrm{E}\left[\lambda_t^{\mathsf{c,a}}\right],$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\lambda_t^{\mathsf{a,c}}\right] = \frac{\bar{\alpha}^{\mathsf{a,c}}}{K_t}\left(\mathrm{E}\left[\lambda_t^{\mathsf{c,c}}\right] + \mathrm{E}\left[\lambda_t^{\mathsf{c,a}}\right]\right) - \frac{\beta^{\mathsf{a,c}}}{K_t}\mathrm{E}\left[\lambda_t^{\mathsf{a,c}}\right],$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\lambda_t^{\mathsf{a,a}}\right] = \frac{\bar{\alpha}^{\mathsf{a,a}}}{K_t}\left(\mathrm{E}\left[\lambda_t^{\mathsf{a,c}}\right] + \mathrm{E}\left[\lambda_t^{\mathsf{a,a}}\right]\right) - \frac{\beta^{\mathsf{a,a}}}{K_t}\mathrm{E}\left[\lambda_t^{\mathsf{a,a}}\right].$$

Letting $\boldsymbol{\lambda}_t = [\lambda_t^{\mathsf{c,c}}, \lambda_t^{\mathsf{c,a}}, \lambda_t^{\mathsf{a,c}}, \lambda_t^{\mathsf{a,a}}]^{\mathrm{T}}$, this pattern of the source of jumps and self decay gives rise to the linear system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\boldsymbol{\lambda}_t\right] = \left(\begin{bmatrix} \bar{\alpha}^{\mathsf{c,c}} & \bar{\alpha}^{\mathsf{c,c}} & & \\ & & \bar{\alpha}^{\mathsf{c,a}} & \bar{\alpha}^{\mathsf{c,a}} \\ \frac{\bar{\alpha}^{\mathsf{a,c}}}{K_t} & \frac{\bar{\alpha}^{\mathsf{a,c}}}{K_t} & & \\ & & \frac{\bar{\alpha}^{\mathsf{a,a}}}{K_t} & \frac{\bar{\alpha}^{\mathsf{a,a}}}{K_t} \end{bmatrix} - \begin{bmatrix} \beta^{\mathsf{c,c}} & & & \\ & \beta^{\mathsf{c,a}} & & \\ & & \frac{\beta^{\mathsf{a,c}}}{K_t} & \\ & & & \frac{\beta^{\mathsf{a,a}}}{K_t} \end{bmatrix}\right)\mathrm{E}\left[\boldsymbol{\lambda}_t\right] = \mathbf{M}\mathrm{E}\left[\boldsymbol{\lambda}_t\right],$$

thus for $\boldsymbol{\lambda}_{t_0} = [\lambda_{t_0}^{\mathsf{c,c}}, \lambda_{t_0}^{\mathsf{c,a}}, \lambda_{t_0}^{\mathsf{a,c}}, \lambda_{t_0}^{\mathsf{a,a}}]^{\mathrm{T}}$ as the vector of known initial values, the vector of mean intensities at time t is given by the solution

$$\mathrm{E}\left[\boldsymbol{\lambda}_t\right] = e^{\mathbf{M}(t-t_0)}\boldsymbol{\lambda}_{t_0}. \tag{19}$$

Now, let us define the corresponding sub-counting processes for the number of messages sent by time t, collected in the column vector $\mathbf{N}_t = [N_t^{\mathsf{c,c}}, N_t^{\mathsf{c,a}}, N_t^{\mathsf{a,c}}, N_t^{\mathsf{a,a}}]^{\mathrm{T}}$. Again, through the infinitesimal generator for these Markov processes, we can see that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\mathbf{N}_t\right] = \mathrm{E}\left[\boldsymbol{\lambda}_t\right] = e^{\mathbf{M}t}\boldsymbol{\lambda}_{t_0},$$

thus we have that the mean counting process vector is given by

$$\mathrm{E}\left[\mathbf{N}_t\right] = \mathbf{N}_{t_0} + \mathbf{M}^{-1}\left(e^{\mathbf{M}(t-t_0)} - \mathbf{I}\right)\boldsymbol{\lambda}_{t_0}. \tag{20}$$

□

We can now make use of the details of this proof in establishing the expected total number of messages.

A.3. Proof of Proposition 2

Proof. As $t \to \infty$ in (20), we can observe that

$$\mathbf{M}\mathrm{E}\left[\mathbf{N}_\infty - \mathbf{N}_{t_0} \mid \boldsymbol{\lambda}_{t_0}\right] = -\boldsymbol{\lambda}_{t_0},$$

So, for each $i, j \in \{\mathsf{c}, \mathsf{a}\}$, this implies that

$$\mathrm{E}\left[N_\infty^{i,j} - N_{t_0}^{i,j} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\lambda_0^{i,j}}{\beta^{i,j}} + \frac{\bar{\alpha}^{i,j}}{\beta^{i,j}}\left(\mathrm{E}\left[N_\infty^{j,\mathsf{c}} - N_{t_0}^{j,\mathsf{c}} \mid \boldsymbol{\lambda}_{t_0}\right] + \mathrm{E}\left[N_\infty^{j,\mathsf{a}} - N_{t_0}^{j,\mathsf{a}} \mid \boldsymbol{\lambda}_{t_0}\right]\right).$$

By noting that the total number of messages sent by one party is the sum of the number they send to the other party and the number they send in follow-up to their own writings, i.e., $\mathrm{E}\left[N_\infty^{i} - N_{t_0}^{i} \mid \boldsymbol{\lambda}_{t_0}\right] = \mathrm{E}\left[N_\infty^{i,\mathsf{c}} - N_{t_0}^{i,\mathsf{c}} \mid \boldsymbol{\lambda}_{t_0}\right] + \mathrm{E}\left[N_\infty^{i,\mathsf{a}} - N_{t_0}^{i,\mathsf{a}} \mid \boldsymbol{\lambda}_{t_0}\right]$, this can be re-expressed as

$$\mathrm{E}\left[N_\infty^{i} - N_{t_0}^{i} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\lambda_0^{i,\mathsf{c}}}{\beta^{i,\mathsf{c}}} + \frac{\lambda_0^{i,\mathsf{a}}}{\beta^{i,\mathsf{a}}} + \frac{\bar{\alpha}^{i,\mathsf{c}}}{\beta^{i,\mathsf{c}}}\mathrm{E}\left[N_\infty^{\mathsf{c}} - N_{t_0}^{\mathsf{c}} \mid \boldsymbol{\lambda}_{t_0}\right] + \frac{\bar{\alpha}^{i,\mathsf{a}}}{\beta^{i,\mathsf{a}}}\mathrm{E}\left[N_\infty^{\mathsf{a}} - N_{t_0}^{\mathsf{a}} \mid \boldsymbol{\lambda}_{t_0}\right],$$

or simply

$$\left(1 - \frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\mathrm{E}\left[N_\infty^{\mathsf{c}} - N_{t_0}^{\mathsf{c}} \mid \boldsymbol{\lambda}_{t_0}\right] = \frac{\lambda_0^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} + \frac{\lambda_0^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} + \frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\mathrm{E}\left[N_\infty^{\mathsf{a}} - N_{t_0}^{\mathsf{a}} \mid \boldsymbol{\lambda}_{t_0}\right],$$

and

$$\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)\mathrm{E}\left[N_\infty^{\mathsf{a}}-N_{t_0}^{\mathsf{a}}\mid\boldsymbol{\lambda}_{t_0}\right]=\frac{\lambda_0^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}+\frac{\lambda_0^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}+\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\mathrm{E}\left[N_\infty^{\mathsf{c}}-N_{t_0}^{\mathsf{c}}\mid\boldsymbol{\lambda}_{t_0}\right].$$

Now, substituting the agent's mean message count into the customer equation, we have

$$\left(\left(1-\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)-\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)\mathrm{E}\left[N_\infty^{\mathsf{c}}-N_{t_0}^{\mathsf{c}}\mid\boldsymbol{\lambda}_{t_0}\right]=\left(\frac{\lambda_0^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}+\frac{\lambda_0^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right)\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)+\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\left(\frac{\lambda_0^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}+\frac{\lambda_0^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right),$$

yielding

$$\mathrm{E}\left[N_\infty^{\mathsf{c}}-N_{t_0}^{\mathsf{c}}\mid\boldsymbol{\lambda}_{t_0}\right]=\frac{\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)\left(\frac{\lambda_0^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}+\frac{\lambda_0^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right)+\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\left(\frac{\lambda_0^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}+\frac{\lambda_0^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)}{\left(1-\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)-\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}},$$

and

$$\mathrm{E}\left[N_\infty^{\mathsf{a}}-N_{t_0}^{\mathsf{a}}\mid\boldsymbol{\lambda}_{t_0}\right]=\frac{\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(\frac{\lambda_0^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}+\frac{\lambda_0^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\right)+\left(1-\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(\frac{\lambda_0^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}+\frac{\lambda_0^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\right)}{\left(1-\frac{\bar{\alpha}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\right)\left(1-\frac{\bar{\alpha}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\right)-\frac{\bar{\alpha}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\frac{\bar{\alpha}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}}.$$

By adding and simplifying these two expressions, we achieve the stated result for the total mean number of messages. ☐

A.4. Proof of Proposition 3

Proof. Note that the event that $S \geq x$ is equivalent to the event that there are no arrivals in $[t_0, t_0 + x)$, i.e., $N_{t_0+x} - N_{t_0} = 0$. Because Hawkes processes are conditionally non-stationary Poisson until the next arrival given the process history, this is

$$\begin{aligned}
\mathrm{P}\left(N_{t_0+x}-N_{t_0}=0\mid\boldsymbol{\lambda}_{t_0}\right)&=e^{-\int_{t_0}^{t_0+x}(\lambda_t^{\mathsf{c}}+\lambda_t^{\mathsf{a}})\mathrm{d}t}\\
&=e^{-\int_0^x\left(\lambda_{t_0}^{\mathsf{c,c}}e^{-\beta^{\mathsf{c,c}}t}+\lambda_{t_0}^{\mathsf{c,a}}e^{-\beta^{\mathsf{c,a}}t}+\frac{\lambda_{t_0}^{\mathsf{a,c}}}{K_{t_0}}e^{-\frac{\beta^{\mathsf{a,c}}t}{K_{t_0}}}+\frac{\lambda_{t_0}^{\mathsf{a,a}}}{K_{t_0}}e^{-\frac{\beta^{\mathsf{a,a}}t}{K_{t_0}}}\right)\mathrm{d}t}\\
&=e^{-\frac{\lambda_{t_0}^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}}\left(1-e^{-\beta^{\mathsf{c,c}}x}\right)-\frac{\lambda_{t_0}^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}}\left(1-e^{-\beta^{\mathsf{c,a}}x}\right)-\frac{\lambda_{t_0}^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}}\left(1-e^{-\frac{\beta^{\mathsf{a,c}}x}{K_{t_0}}}\right)-\frac{\lambda_{t_0}^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}}\left(1-e^{-\frac{\beta^{\mathsf{a,a}}x}{K_{t_0}}}\right)}.
\end{aligned}$$

Taking the limit as $x \to \infty$, we complete the proof. ☐

B. Description of Estimation and Simulation Procedures

B.1. Estimation Algorithm

To estimate the parameters of these processes from data, we use a variant of the expectation-maximization (EM) algorithm. As we will describe in this section, these procedures are highly efficient and easily implementable in practice and are thus quite common in the Hawkes process literature. Examples of successfully applying these algorithms to self-exciting processes can be seen in works such as Lewis and Mohler (2011), Halpin (2012), Halpin and De Boeck (2013). It is worth noting that each of these are themselves an extension of the general branching process EM approach developed in Veen and Schoenberg (2008), as is the algorithm we give in this section. The tractability of these algorithms for Markovian Hawkes processes largely lies in the fact that all the supporting calculations within each iteration reduce to solving simple linear equations, which can be found in closed form. In this way, it can be shown that this EM algorithm is equivalent to projected gradient ascent on the log-likelihood function (Lewis and Mohler 2011). Nevertheless, other methods of estimation exist in the literature for Hawkes processes, so let us briefly mention a few alternatives. First, the most comparable procedure is maximum likelihood estimation (MLE), as the EM algorithm also relies on the likelihood function. This function was first provided in Ozaki (1979). While EM algorithms do not necessarily have the same level of theoretical guarantees, they do offer considerable computational advantages over the non-linear optimization of direct maximum likelihood estimation on large datasets such as the one we study in this work. By comparison, one could instead use parametric approaches that draw upon advanced optimization techniques, such as in Guo et al. (2018). There are also interesting approaches available for the alternate setting in which there are only a small number of data points available, e.g., in Salehi et al. (2019). For an overview and comparison of Hawkes process estimation procedures, see Kirchner and Bercher (2018).

As we have noted, we use the EM algorithm because of its computational simplicity and ease of implementation. This tractability means we can also easily describe the EM approach. Although all the terms can be written in closed form, some become cumbersome. Hence, we reserve some explicit computations for Appendix B.2. Because the three models encapsulate one another, we only describe the estimation procedure for the SysBHP model in detail. The other models can be simplified from this. Just as the conversations are the focal point of our modeling approach, this structure will also be key to our estimation procedure. In particular, because it is known which messages belong to which conversation, the estimation calculations can be distributed across each messaging conversation. Data from one conversation can be considered separately from all other conversations, and this follows from the independence of branches within the Hawkes process models. The data points we use are the message time stamps and sending parties, meaning the customer or agent. Because system features like the concurrency, sentiment scores, and numbers of words per message are observable in practice, we treat these quantities as known. Hence, we only seek to estimate the jump size and decay parameters.

To describe the EM algorithm, we begin by first specifying the log-likelihood function for a given conversation. Because the Hawkes process models are stochastic intensity Poisson processes, we can give the log-likelihood in closed form. In the case of the SysBHP conversation model, this is given by

$$\mathcal{L}(\theta \mid \mathcal{D}) = \sum_{i=1}^{N_\infty^{\mathsf{c}}} \log\left(\lambda^{\mathsf{c}}_{A_i^{\mathsf{c}}-}\right) + \sum_{j=1}^{N_\infty^{\mathsf{a}}} \log\left(\lambda^{\mathsf{a}}_{A_j^{\mathsf{a}}-}\right) - \int_0^\infty \lambda_t^{\mathsf{c}} \mathrm{d}t - \int_0^\infty \lambda_t^{\mathsf{a}} \mathrm{d}t, \tag{21}$$

where λ_t^{c} and λ_t^{a} are respectively the customer and agent correspondence rates given in Equations (6) and (7) with $\lambda^{\mathsf{c}}_{A_i^{\mathsf{c}}-} = \lim_{t\uparrow A_i^c} \lambda_t^{\mathsf{c}}$ and $\lambda^{\mathsf{a}}_{A_j^{\mathsf{a}}-} = \lim_{t\uparrow A_j^a} \lambda_t^{\mathsf{a}}$, where $\theta = \{\alpha_1^{\mathsf{c,c}}, \alpha_1^{\mathsf{c,a}}, \alpha_1^{\mathsf{a,c}}, \alpha_1^{\mathsf{a,a}}, \alpha_2^{\mathsf{c,c}}, \alpha_2^{\mathsf{c,a}}, \alpha_2^{\mathsf{a,c}}, \alpha_2^{\mathsf{a,a}}, \beta^{\mathsf{c,c}}, \beta^{\mathsf{c,a}}, \beta^{\mathsf{a,c}}, \beta^{\mathsf{a,a}}\}$ is the parameter set, and where $\mathcal{D} = \{(A_1^{\mathsf{c}}, \dots, A_{N^{\mathsf{c}}}^{\mathsf{c}}), (A_1^{\mathsf{a}}, \dots, A_{N^{\mathsf{a}}}^{\mathsf{a}})\}$ is the message timestamps data set for the full conversation. Because the sentiments and word counts are drawn independently from the correspondence processes, we do not include their distributions in the likelihood expressions; these terms will vanish when taking partial derivatives of $\mathcal{L}$ with respect to the parameters in θ. The fully simplified log-likelihood is given in Appendix B.2. Note that because the data comprises only completed conversations and all conversations contain finitely many messages, we are using $N^{\mathsf{c}}_\infty = \lim_{t\to\infty} N^{\mathsf{c}}_t$ and $N^{\mathsf{a}}_\infty = \lim_{t\to\infty} N^{\mathsf{a}}_t$ as the total number of customer and agent messages in the conversation, excluding the initial query. Because conversations are conditionally independent from one another given the concurrency of the agents, we can then note that the log-likelihood function of the full contact center data containing $M \in \mathbb{Z}^+$ conversations, say $\bar{\mathcal{L}}(\theta \mid \bar{\mathcal{D}})$, can be obtained from

$$\bar{\mathcal{L}}(\theta \mid \bar{\mathcal{D}}) = \sum_{m=1}^{M} \mathcal{L}_m \left(\theta \mid \mathcal{D}_m\right),$$

where $\mathcal{L}_m \left(\theta \mid \mathcal{D}_m\right)$ is the log-likelihood for the m^{th} conversation as calculated according to Equation (24) and where $\bar{\mathcal{D}} = \bigcup_{m=1}^{M} \mathcal{D}_m$ is the complete data set.

EM algorithms work by making use of missing data. In our setting, the missing data is the precise conversational dependencies, meaning knowledge of which previous message prompted a given message as response. This is not observable in the data, but we can quantify the probability that one message is in response to another. For example, given the parameters of the SysBHP conversation model and the conversation data, the probability that the i^{th} customer message is actually in response to j^{th} customer message and is spurred by the sentiment of this message can be calculated via

$$p^{\mathsf{c},\mathsf{c}-1}_{i,j} = \frac{1}{\lambda^{\mathsf{c}}_{A^{\mathsf{c}}_i-}} \alpha^{\mathsf{c},\mathsf{c}}_1 S^{\mathsf{c}}_j e^{-\beta^{\mathsf{c},\mathsf{c}}(A^{\mathsf{c}}_i - A^{\mathsf{c}}_j)}, \tag{22}$$

since this is the amount of excitement generated by the j^{th} customer message within the customer message intensity at the time the i^{th} customer message was sent. Likewise, the probability that the i^{th} message is spurred by the word count of the j^{th} message is

$$p^{\mathsf{c},\mathsf{c}-2}_{i,j} = \frac{1}{\lambda^{\mathsf{c}}_{A^{\mathsf{c}}_i-}} \alpha^{\mathsf{c},\mathsf{c}}_2 W^{\mathsf{c}}_j e^{-\beta^{\mathsf{c},\mathsf{c}}(A^{\mathsf{c}}_i - A^{\mathsf{c}}_j)},$$

Similarly, the other response probabilities can thus be calculated as

$$\begin{aligned}
p^{\mathsf{c},\mathsf{a}-1}_{i,j} &= \frac{1}{\lambda^{\mathsf{c}}_{A^{\mathsf{c}}_i-}} \alpha^{\mathsf{c},\mathsf{a}}_1 S^{\mathsf{a}}_j e^{-\beta^{\mathsf{c},\mathsf{a}}(A^{\mathsf{c}}_i - A^{\mathsf{a}}_j)}, \qquad && p^{\mathsf{c},\mathsf{a}-2}_{i,j} = \frac{1}{\lambda^{\mathsf{c}}_{A^{\mathsf{c}}_i-}} \alpha^{\mathsf{c},\mathsf{a}}_2 W^{\mathsf{a}}_j e^{-\beta^{\mathsf{c},\mathsf{a}}(A^{\mathsf{c}}_i - A^{\mathsf{a}}_j)}, \\
p^{\mathsf{a},\mathsf{c}-1}_{i,j} &= \frac{1}{\lambda^{\mathsf{a}}_{A^{\mathsf{a}}_i-} K_{A^{\mathsf{a}}_i}} \alpha^{\mathsf{a},\mathsf{c}}_1 S^{\mathsf{c}}_j e^{-\beta^{\mathsf{a},\mathsf{c}}(A^{\mathsf{a}}_i - A^{\mathsf{c}}_j)/K_{A^{\mathsf{a}}_i}}, \qquad && p^{\mathsf{a},\mathsf{c}-2}_{i,j} = \frac{1}{\lambda^{\mathsf{a}}_{A^{\mathsf{a}}_i-} K_{A^{\mathsf{a}}_i}} \alpha^{\mathsf{a},\mathsf{c}}_2 W^{\mathsf{c}}_j e^{-\beta^{\mathsf{a},\mathsf{c}}(A^{\mathsf{a}}_i - A^{\mathsf{c}}_j)/K_{A^{\mathsf{a}}_i}}, \\
p^{\mathsf{a},\mathsf{a}-1}_{i,j} &= \frac{1}{\lambda^{\mathsf{a}}_{A^{\mathsf{a}}_i-} K_{A^{\mathsf{a}}_i}} \alpha^{\mathsf{a},\mathsf{a}}_1 S^{\mathsf{a}}_j e^{-\beta^{\mathsf{a},\mathsf{a}}(A^{\mathsf{a}}_i - A^{\mathsf{a}}_j)/K_{A^{\mathsf{a}}_i}}, \text{ and } && p^{\mathsf{a},\mathsf{a}-2}_{i,j} = \frac{1}{\lambda^{\mathsf{a}}_{A^{\mathsf{a}}_i-} K_{A^{\mathsf{a}}_i}} \alpha^{\mathsf{a},\mathsf{a}}_2 W^{\mathsf{a}}_j e^{-\beta^{\mathsf{a},\mathsf{a}}(A^{\mathsf{a}}_i - A^{\mathsf{a}}_j)/K_{A^{\mathsf{a}}_i}}.
\end{aligned} \tag{23}$$

Given these response probabilities, one can also then calculate the value of the parameters that are critical points for the full system log-likelihood. By first re-parameterizing the jump sizes in proportion to the decay rate, i.e. $\hat{\alpha}^{\mathsf{c},\mathsf{c}}_1 = \frac{\alpha^{\mathsf{c},\mathsf{c}}_1}{\beta^{\mathsf{c},\mathsf{c}}}$, one can in fact give these parameter solutions in closed form, as this change of variable yields that the critical point of each partial derivative is simply found through solving a linear equation. Of course, upon completion of the EM algorithm, one can then obtain the true model jump sizes by simply multiplying $\hat{\alpha}$ by β. Because of their length, these expressions are available in the next subsection, Appendix B.2, so as to not distract from the overarching ideas. This pair of calculations gives us the basis of the iterative EM algorithm, for which we now provide pseudocode in Algorithm 1.

Recall from the main body of the text that the `E-step` of Algorithm 1 has an important subtlety. By comparison to traditional estimation settings of the Hawkes process, we have a significant advantage in

Algorithm 1: The SysBHP EM Algorithm

Result: Jump sizes $\vec{\alpha}_*^{(t)}$ and decay rates $\vec{\beta}_*^{(t)}$.

Initialization: Choose the starting parameters $\vec{\alpha}_*^{(0)}$ and $\vec{\beta}_*^{(0)}$ randomly.

while $||\vec{\alpha}_*^{(t)} - \vec{\alpha}_*^{(t-1)}|| + ||\vec{\beta}_*^{(t)} - \vec{\beta}_*^{(t-1)}|| > \epsilon$ **do**

`E-step`: Given the observed data and current parameter estimates $\vec{\alpha}_*^{(t)}$ and $\vec{\beta}_*^{(t)}$, compute the updated response probabilities (each $p_{i,j}^{\mathsf{c,c}-1}$, $p_{i,j}^{\mathsf{c,a}-1}$, $p_{i,j}^{\mathsf{a,c}-1}$, $p_{i,j}^{\mathsf{a,a}-1}$, $p_{i,j}^{\mathsf{c,c}-2}$, $p_{i,j}^{\mathsf{c,a}-2}$, $p_{i,j}^{\mathsf{a,c}-2}$, and $p_{i,j}^{\mathsf{a,a}-2}$) within each conversation through Equations (22) and (23).

`M-step`: Using the newly calculated response probabilities and the previous parameter estimates, compute the new parameter estimates $\vec{\alpha}_*^{(t+1)}$ and $\vec{\beta}_*^{(t+1)}$ as the solutions to the linear critical point equations, as given in Equations (25) through (32).

$t \leftarrow t+1$.

end

complexity by knowing which messages lie within each conversation. In terms of the Hawkes processes, this means that we know which branch each event occurred in, even if we do not know the specific ordering within the branch. This gives us an EM algorithm that is $O\Big(\sum_{m=1}^{M} N_m^2\Big)$ rather than $O\Big(\big(\sum_{m=1}^{M} N_m\big)^2\Big)$ where N_m is the total number of messages in conversation m, since we do not need to check whether a given message is in response to any message from a different conversation. On a data set with this many distinct conversations, this is a substantial simplification, even by comparison to standard Hawkes EM implementations.

B.2. Log-Likelihood and EM Algorithm Equations

Here we derive the full log-likelihood for the SysBHP model; the other models can be simplified from this. Following substitution and simplification from the definition of the correspondence rates and the representation of the log-likelihood in Equation (21), this log-likelihood can also be expressed

$$
\begin{aligned}
\mathcal{L}(\theta \mid \mathcal{D}) = &\sum_{k=1}^{N^{\mathsf{c}}} \log \left(\sum_{i=0}^{k-1} (\alpha_1^{\mathsf{c,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{c,c}} W_i^{\mathsf{c}})\, e^{-\beta^{\mathsf{c,c}}(A_k^{\mathsf{c}} - A_i^{\mathsf{c}})} + \sum_{j=1}^{N^{\mathsf{a}}_{A_k^{\mathsf{c}}}} (\alpha_1^{\mathsf{c,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{c,a}} W_j^{\mathsf{a}})\, e^{-\beta^{\mathsf{c,a}}(A_k^{\mathsf{c}} - A_j^{\mathsf{a}})} \right) \qquad (24)\\
&+ \sum_{k=1}^{N^{\mathsf{a}}} \log \left(\sum_{i=0}^{N^{\mathsf{c}}_{A_k^{\mathsf{a}}}} \frac{\alpha_1^{\mathsf{a,c}} S_i^{\mathsf{c}} + \alpha_2^{\mathsf{a,c}} W_i^{\mathsf{c}}}{K_{A_k^{\mathsf{a}}}} e^{-\beta^{\mathsf{a,c}}(A_k^{\mathsf{a}} - A_i^{\mathsf{c}})/K_{A_k^{\mathsf{a}}}} + \sum_{j=1}^{k-1} \frac{\alpha_1^{\mathsf{a,a}} S_j^{\mathsf{a}} + \alpha_2^{\mathsf{a,a}} W_j^{\mathsf{a}}}{K_{A_k^{\mathsf{a}}}} e^{-\beta^{\mathsf{a,a}}(A_k^{\mathsf{a}} - A_j^{\mathsf{a}})/K_{A_k^{\mathsf{a}}}} \right)\\
&- \sum_{i=0}^{N^{\mathsf{c}}} \left(\frac{\alpha_1^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} S_i^{\mathsf{c}} + \frac{\alpha_2^{\mathsf{c,c}}}{\beta^{\mathsf{c,c}}} W_i^{\mathsf{c}} \right) - \sum_{i=0}^{N^{\mathsf{c}}} \left(\frac{\alpha_1^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} S_i^{\mathsf{c}} + \frac{\alpha_2^{\mathsf{a,c}}}{\beta^{\mathsf{a,c}}} W_i^{\mathsf{c}} \right) \sum_{k=1}^{\kappa} \left(e^{-\beta^{\mathsf{a,c}} f(K_{\Delta_{k-1}})(\Delta_{k-1} - A_i^{\mathsf{c}})^+} - e^{-\beta^{\mathsf{a,c}} f(K_{\Delta_{k-1}})(\Delta_k - A_i^{\mathsf{c}})^+} \right)\\
&- \sum_{j=1}^{N^{\mathsf{a}}} \left(\frac{\alpha_1^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} S_j^{\mathsf{a}} + \frac{\alpha_2^{\mathsf{c,a}}}{\beta^{\mathsf{c,a}}} W_j^{\mathsf{a}} \right) - \sum_{j=1}^{N^{\mathsf{a}}} \left(\frac{\alpha_1^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} S_j^{\mathsf{a}} + \frac{\alpha_2^{\mathsf{a,a}}}{\beta^{\mathsf{a,a}}} W_j^{\mathsf{a}} \right) \sum_{k=1}^{\kappa} \left(e^{-\beta^{\mathsf{a,a}} f(K_{\Delta_{k-1}})(\Delta_{k-1} - A_j^{\mathsf{a}})^+} - e^{-\beta^{\mathsf{a,a}} f(K_{\Delta_{k-1}})(\Delta_k - A_j^{\mathsf{a}})^+} \right),
\end{aligned}
$$

where κ is the total number of successive concurrency values that occurred over the course of this conversation, with $K_t = K_{\Delta_{k-1}}$ on $t \in [\Delta_{k-1}, \Delta_k)$ for each $k \leq \kappa$.

Taking the subscript $*$ for the roots of the first derivative of the log-likelihood with respect to each parameter, we can express the jump sizes in terms of the responses probabilities as

$$
\hat{\alpha}_{1,*}^{\mathsf{c,c}} = \frac{\sum_{m=1}^{M} \sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}} \sum_{i=0}^{k-1} p_{k,i,m}^{\mathsf{c,c}-1}}{\sum_{m=1}^{M} \sum_{i=0}^{N^{\mathsf{c}}_{\infty,m}} S_{i,m}^{\mathsf{c}}}, \quad \hat{\alpha}_{2,*}^{\mathsf{c,c}} = \frac{\sum_{m=1}^{M} \sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}} \sum_{i=0}^{k-1} p_{k,i,m}^{\mathsf{c,c}-2}}{\sum_{m=1}^{M} \sum_{i=0}^{N^{\mathsf{c}}_{\infty,m}} W_{i,m}^{\mathsf{c}}}, \qquad (25)
$$

$$\hat{\alpha}_{1,*}^{\mathsf{c},\mathsf{a}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{j=1}^{N^{\mathsf{a}}_{A^{\mathsf{c}}_{k,m},m}} p_{k,j,m}^{\mathsf{c},\mathsf{a}-1}}{\sum_{m=1}^{M}\sum_{j=1}^{N^{\mathsf{a}}_{\infty,m}} S^{\mathsf{a}}_{j,m}}, \quad \hat{\alpha}_{2,*}^{\mathsf{c},\mathsf{a}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{j=1}^{N^{\mathsf{a}}_{A^{\mathsf{c}}_{k,m},m}} p_{k,j,m}^{\mathsf{c},\mathsf{a}-2}}{\sum_{m=1}^{M}\sum_{j=1}^{N^{\mathsf{a}}_{\infty,m}} W^{\mathsf{a}}_{j,m}}, \tag{26}$$

$$\hat{\alpha}_{1,*}^{\mathsf{a},\mathsf{c}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{i=0}^{N^{\mathsf{c}}_{A^{\mathsf{a}}_{k,m},m}} p_{k,i,m}^{\mathsf{a},\mathsf{c}-1}}{\sum_{m=1}^{M}\sum_{i=0}^{N^{\mathsf{c}}_{\infty,m}} S^{\mathsf{c}}_{i,m}\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{c}}/K_{\Delta_{k-1,m}}(\Delta_{k-1,m}-A^{\mathsf{c}}_{i,m})^+} - e^{-\beta^{\mathsf{a},\mathsf{c}}/K_{\Delta_{k-1,m}}(\Delta_{k,m}-A^{\mathsf{c}}_{i,m})^+}\right)},$$

$$\hat{\alpha}_{2,*}^{\mathsf{a},\mathsf{c}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{i=0}^{N^{\mathsf{c}}_{A^{\mathsf{a}}_{k,m},m}} p_{k,i,m}^{\mathsf{a},\mathsf{c}-2}}{\sum_{m=1}^{M}\sum_{i=0}^{N^{\mathsf{c}}_{\infty,m}} W^{\mathsf{c}}_{i,m}\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{c}}/K_{\Delta_{k-1,m}}(\Delta_{k-1,m}-A^{\mathsf{c}}_{i,m})^+} - e^{-\beta^{\mathsf{a},\mathsf{c}}/K_{\Delta_{k-1,m}}(\Delta_{k,m}-A^{\mathsf{c}}_{i,m})^+}\right)}, \tag{27}$$

and

$$\hat{\alpha}_{1,*}^{\mathsf{a},\mathsf{a}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{j=1}^{k-1} p_{k,j,m}^{\mathsf{a},\mathsf{a}-1}}{\sum_{m=1}^{M}\sum_{j=1}^{N^{\mathsf{a}}_{\infty,m}} S^{\mathsf{a}}_{j,m}\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{a}}/K_{\Delta_{k-1,m}}(\Delta_{k-1,m}-A^{\mathsf{a}}_{j,m})^+} - e^{-\beta^{\mathsf{a},\mathsf{a}}/K_{\Delta_{k-1,m}}(\Delta_{k,m}-A^{\mathsf{a}}_{j,m})^+}\right)},$$

$$\hat{\alpha}_{2,*}^{\mathsf{a},\mathsf{a}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{j=1}^{k-1} p_{k,j,m}^{\mathsf{a},\mathsf{a}-2}}{\sum_{m=1}^{M}\sum_{j=1}^{N^{\mathsf{a}}_{\infty,m}} W^{\mathsf{a}}_{j,m}\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{a}}/K_{\Delta_{k-1,m}}(\Delta_{k-1,m}-A^{\mathsf{a}}_{j,m})^+} - e^{-\beta^{\mathsf{a},\mathsf{a}}/K_{\Delta_{k-1,m}}(\Delta_{k,m}-A^{\mathsf{a}}_{j,m})^+}\right)}. \tag{28}$$

Likewise, the decay rates are given by

$$\beta_*^{\mathsf{c},\mathsf{c}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{i=0}^{k-1}(p_{k,i,m}^{\mathsf{c},\mathsf{c}-1}+p_{k,i,m}^{\mathsf{c},\mathsf{c}-2})}{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{i=0}^{k-1}(p_{k,i,m}^{\mathsf{c},\mathsf{c}-1}+p_{k,i,m}^{\mathsf{c},\mathsf{c}-2})\left(A^{\mathsf{c}}_{k,m}-A^{\mathsf{c}}_{i,m}\right)}, \tag{29}$$

$$\beta_*^{\mathsf{c},\mathsf{a}} = \frac{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{j=1}^{N^{\mathsf{a}}_{A^{\mathsf{c}}_{k,m},m}}(p_{k,j,m}^{\mathsf{c},\mathsf{a}-1}+p_{k,j,m}^{\mathsf{c},\mathsf{a}-2})}{\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{c}}_{\infty,m}}\sum_{j=1}^{N^{\mathsf{a}}_{A^{\mathsf{c}}_{k,m},m}}(p_{k,j,m}^{\mathsf{c},\mathsf{a}-1}+p_{k,j,m}^{\mathsf{c},\mathsf{a}-2})\left(A^{\mathsf{c}}_{k,m}-A^{\mathsf{a}}_{j,m}\right)}, \tag{30}$$

$$\begin{aligned}\beta_*^{\mathsf{a},\mathsf{c}} = &\left(\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{i=0}^{N^{\mathsf{c}}_{A^{\mathsf{a}}_{k,m},m}}(p_{k,i,m}^{\mathsf{a},\mathsf{c}-1}+p_{k,i,m}^{\mathsf{a},\mathsf{c}-2})\right)\Bigg/\left(\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{i=0}^{N^{\mathsf{c}}_{A^{\mathsf{a}}_{k,m},m}}(p_{k,i,m}^{\mathsf{a},\mathsf{c}-1}+p_{k,i,m}^{\mathsf{a},\mathsf{c}-2})\frac{A^{\mathsf{a}}_{k,m}-A^{\mathsf{c}}_{i,m}}{K_{A^{\mathsf{a}}_{k,m}}}\right.\\ &-\sum_{m=1}^{M}\sum_{i=0}^{N^{\mathsf{c}}_{\infty,m}}(\hat{\alpha}_1^{\mathsf{a},\mathsf{c}}S^{\mathsf{c}}_{i,m}+\hat{\alpha}_2^{\mathsf{a},\mathsf{c}}W^{\mathsf{c}}_{i,m})\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{c}}\frac{(\Delta_{k-1,m}-A^{\mathsf{c}}_{i,m})^+}{K_{\Delta_{k-1,m}}}}\frac{(\Delta_{k-1,m}-A^{\mathsf{c}}_{i,m})^+}{K_{\Delta_{k-1,m}}}\right.\\ &\left.\left.-\,e^{-\beta^{\mathsf{a},\mathsf{c}}\frac{(\Delta_{k,m}-A^{\mathsf{c}}_{i,m})^+}{K_{\Delta_{k-1,m}}}}\frac{(\Delta_{k,m}-A^{\mathsf{c}}_{i,m})^+}{K_{\Delta_{k-1,m}}}\right)\right),\end{aligned} \tag{31}$$

and

$$\begin{aligned}\beta_*^{\mathsf{a},\mathsf{a}} = &\left(\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{j=1}^{k-1}(p_{k,j,m}^{\mathsf{a},\mathsf{a}-1}+p_{k,j,m}^{\mathsf{a},\mathsf{a}-2})\right)\Bigg/\left(\sum_{m=1}^{M}\sum_{k=1}^{N^{\mathsf{a}}_{\infty,m}}\sum_{j=1}^{k-1}(p_{k,j,m}^{\mathsf{a},\mathsf{a}-1}+p_{k,j,m}^{\mathsf{a},\mathsf{a}-2})\frac{A^{\mathsf{a}}_{k,m}-A^{\mathsf{a}}_{j,m}}{K_{A^{\mathsf{a}}_{k,m}}}\right.\\ &-\sum_{m=1}^{M}\sum_{j=1}^{N^{\mathsf{a}}_{\infty,m}}(\hat{\alpha}_1^{\mathsf{a},\mathsf{a}}S^{\mathsf{a}}_{j,m}+\hat{\alpha}_2^{\mathsf{a},\mathsf{a}}W^{\mathsf{a}}_{j,m})\sum_{k=1}^{\kappa_m}\left(e^{-\beta^{\mathsf{a},\mathsf{a}}\frac{(\Delta_{k-1,m}-A^{\mathsf{a}}_{j,m})^+}{K_{\Delta_{k-1,m}}}}\frac{(\Delta_{k-1,m}-A^{\mathsf{a}}_{j,m})^+}{K_{\Delta_{k-1,m}}}\right.\\ &\left.\left.-\,e^{-\beta^{\mathsf{a},\mathsf{a}}\frac{(\Delta_{k,m}-A^{\mathsf{a}}_{j,m})^+}{K_{\Delta_{k-1,m}}}}\frac{(\Delta_{k,m}-A^{\mathsf{a}}_{j,m})^+}{K_{\Delta_{k-1,m}}}\right)\right).\end{aligned} \tag{32}$$

With these quantities in hand, one can directly compute all steps of Algorithm 1.

C. Model Evaluation: Gap Times, Number of Gaps, and Parameters for SED model

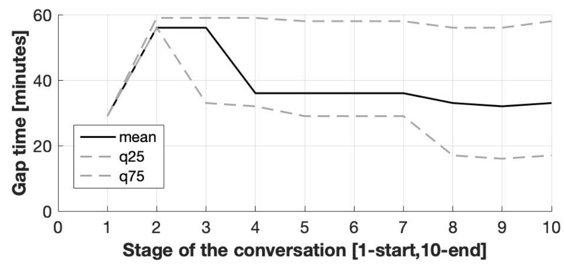


Figure 8 Gap time as a function of the stage of the conversation, May 2017.

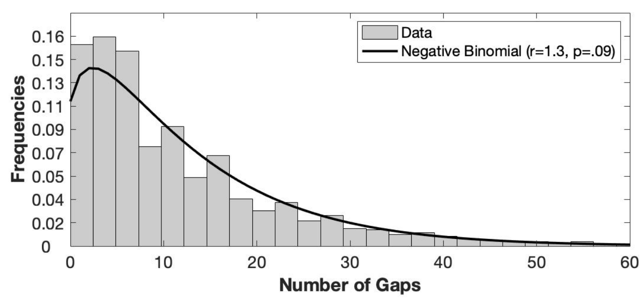


Figure 9 Negative binomial distribution fit on number of gaps.

Table 3 Estimated Parameters for SED Model (Training set May 1-23 2017).

Gap number	$1/\mu$
1	$1/\mu_1 = 0.183$
2	$1/\mu_2 = 0.085$
3	$1/\mu_3 = 0.095$
4	$1/\mu_4 = 0.092$
5	$1/\mu_5 = 0.081$
6	$1/\mu_6 = 0.073$
7	$1/\mu_7 = 0.065$
8	$1/\mu_8 = 0.062$
9	$1/\mu_9 = 0.059$
10	$1/\mu_{10} = 0.058$
11	$1/\mu_{11} = 0.056$
12	$1/\mu_{12} = 0.054$
13	$1/\mu_{13} = 0.052$
>14	$1/\mu_{>14} = 0.046$

Table 4 Mean Inner Wait (mins) for the Lightest Load Routing Policies with and without Hawkes Projections.

		$\delta = 0.5$ Minutes				**$\delta = 5$ Minutes**				**$\delta = \infty$ (End of Conv.)**			
$\mathcal{S}$	κ	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP
125	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	3.9	3.7 (3.8%)	3.7 (4.6%)	3.7 (4.1%)	3.9	3.7 (3.8%)	3.7 (4.3%)	3.7 (4.7%)	3.9	3.7 (3.7%)	3.7 (4.4%)	3.7 (3.2%)
	10	9.5	9.1 (3.6%)	9.0 (4.7%)	9.0 (4.9%)	9.6	9.1 (4.5%)	9.1 (4.9%)	9.0 (5.3%)	9.5	9.1 (4.1%)	9.1 (4.4%)	9.1 (4.0%)
	15	14.3	13.7 (3.8%)	13.7 (4.3%)	13.6 (4.8%)	14.3	13.7 (3.7%)	13.7 (4.1%)	13.6 (4.6%)	14.3	13.7 (3.6%)	13.7 (3.7%)	13.7 (3.8%)
	20	17.4	16.8 (3.5%)	16.7 (4.1%)	16.6 (4.9%)	17.5	16.8 (3.9%)	16.8 (4.1%)	16.6 (5.3%)	17.4	16.8 (3.3%)	16.9 (2.7%)	16.7 (3.9%)
	25	19.0	18.4 (3.0%)	18.2 (4.1%)	18.1 (4.7%)	19.0	18.4 (2.8%)	18.4 (3.0%)	18.2 (4.1%)	19.1	18.4 (3.3%)	18.5 (2.7%)	18.3 (3.9%)
	30	19.7	19.0 (3.6%)	18.8 (4.6%)	18.6 (5.5%)	19.5	19.0 (2.5%)	19.1 (2.0%)	18.7 (3.9%)	19.5	19.0 (2.6%)	19.1 (1.8%)	18.9 (3.1%)
135	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	3.8	3.6 (4.1%)	3.6 (4.4%)	3.6 (4.6%)	3.8	3.6 (4.7%)	3.6 (5.6%)	3.6 (4.7%)	3.8	3.6 (3.7%)	3.6 (4.1%)	3.6 (3.8%)
	10	8.9	8.4 (5.0%)	8.3 (6.1%)	8.3 (6.1%)	8.9	8.4 (5.3%)	8.4 (5.8%)	8.3 (6.1%)	8.9	8.4 (5.1%)	8.4 (5.3%)	8.4 (5.0%)
	15	13.0	12.4 (4.3%)	12.3 (5.2%)	12.2 (5.6%)	12.8	12.4 (3.4%)	12.3 (4.0%)	12.3 (4.3%)	13.0	12.4 (4.3%)	12.3 (4.8%)	12.3 (5.1%)
	20	15.0	14.5 (3.4%)	14.3 (4.6%)	14.3 (4.8%)	15.1	14.5 (4.1%)	14.3 (5.0%)	14.3 (4.9%)	15.1	14.5 (4.1%)	14.4 (4.8%)	14.3 (5.0%)
	25	15.8	15.2 (3.7%)	15.0 (4.9%)	15.0 (5.0%)	15.8	15.2 (3.5%)	15.1 (4.4%)	15.1 (4.3%)	15.8	15.2 (3.7%)	15.1 (4.6%)	15.0 (5.1%)
	30	15.8	15.3 (3.4%)	15.1 (4.6%)	15.0 (4.8%)	15.9	15.3 (4.1%)	15.1 (5.0%)	15.1 (4.8%)	15.9	15.3 (4.0%)	15.1 (5.0%)	15.0 (5.5%)
145	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	3.7	3.5 (5.0%)	3.5 (5.2%)	3.5 (5.2%)	3.7	3.5 (5.5%)	3.5 (6.1%)	3.5 (5.6%)	3.7	3.5 (5.9%)	3.5 (6.3%)	3.5 (6.1%)
	10	8.2	7.9 (4.7%)	7.7 (6.2%)	7.7 (6.2%)	8.2	7.9 (4.7%)	7.7 (6.2%)	7.8 (5.7%)	8.2	7.9 (4.4%)	7.8 (5.1%)	7.8 (4.6%)
	15	11.3	10.9 (3.3%)	10.8 (4.8%)	10.8 (4.9%)	11.4	10.9 (4.3%)	10.7 (6.5%)	10.8 (5.4%)	11.5	10.9 (4.7%)	10.9 (5.3%)	10.8 (6.1%)
	20	12.7	12.2 (4.1%)	12.0 (5.4%)	11.9 (6.1%)	12.7	12.2 (3.7%)	11.9 (5.9%)	12.1 (4.8%)	12.7	12.2 (3.7%)	12.1 (4.2%)	12.0 (5.3%)
	25	12.7	12.3 (2.8%)	12.1 (4.1%)	12.0 (5.1%)	12.8	12.3 (4.2%)	12.0 (6.5%)	12.1 (5.4%)	12.8	12.3 (4.0%)	12.2 (4.7%)	12.1 (5.7%)
	30	12.8	12.3 (4.0%)	12.1 (5.2%)	12.0 (6.2%)	12.8	12.3 (4.0%)	12.0 (6.3%)	12.1 (5.2%)	12.9	12.3 (4.5%)	12.2 (5.2%)	12.1 (6.2%)

The percent improvement is given in parenthesis. All differences in mean are statistically significant (using t-tests) except for $\kappa = 1$.

Table 5 Mean Outer Wait (mins) for the Lightest Load Routing Policies with and without Hawkes Projections.

		$\delta = 0.5$ Minutes				**$\delta = 5$ Minutes**				**$\delta = \infty$ (End of Conv.)**			
$\mathcal{S}$	κ	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP
125	1	2598.9	2598.9 (0.0%)	2598.9 (0.0%)	2566.8 (1.2%)	2598.9	2598.9 (0.0%)	2598.9 (0.0%)	2566.8 (1.2%)	2598.9	2598.9 (0.0%)	2598.9 (0.0%)	2566.8 (1.2%)
	5	125.7	122.4 (2.6%)	122.4 (2.6%)	119.5 (4.9%)	125.9	122.4 (2.8%)	122.1 (3.0%)	119.1 (5.4%)	125.1	122.4 (2.2%)	122.0 (2.5%)	119.9 (4.2%)
	10	25.3	24.6 (2.9%)	23.8 (6.0%)	22.8 (9.9%)	25.8	24.6 (4.8%)	24.2 (6.5%)	23.4 (9.3%)	25.3	24.6 (3.0%)	23.9 (5.8%)	23.4 (7.6%)
	15	11.7	10.8 (7.5%)	10.6 (9.2%)	10.3 (11.7%)	11.5	10.8 (6.0%)	10.8 (5.8%)	10.4 (9.0%)	11.5	10.8 (6.7%)	10.8 (6.0%)	10.4 (10.2%)
	20	4.7	4.1 (12.0%)	4 (14.3%)	3.9 (16.7%)	4.8	4.1 (13.5%)	4.4 (8.5%)	4.1 (15.1%)	4.6	4.1 (10.7%)	4.3 (7.1%)	4.1 (12.0%)
	25	1.8	1.6 (7.1%)	1.5 (15.0%)	1.4 (18.4%)	1.8	1.6 (9.8%)	1.7 (6.8%)	1.5 (19.5%)	1.8	1.6 (8.9%)	1.7 (6.3%)	1.5 (14.6%)
	30	0.2	0.1 (26.9%)	0.1 (12.9%)	0.1 (51.6%)	0.1	0.1 (6.1%)	0.2 (-35.3%)	0.1 (1.2%)	0.2	0.1 (31.6%)	0.1 (37.1%)	0.1 (33.7%)
135	1	2338.6	2338.6 (0.0%)	2338.6 (0.0%)	2308.8 (1.3%)	2338.6	2338.6 (0.0%)	2338.6 (0.0%)	2308.8 (1.3%)	2338.6	2338.6 (0.0%)	2338.6 (0.0%)	2308.8 (1.3%)
	5	97.7	94.9 (2.8%)	94.7 (3%)	92.6 (5.2%)	98.4	94.9 (3.5%)	95.1 (3.4%)	93.0 (5.5%)	97.5	94.9 (2.6%)	95.1 (2.5%)	92.8 (4.8%)
	10	16.1	15.2 (5.7%)	14.8 (8.2%)	14.9 (7.5%)	16.4	15.2 (6.9%)	14.9 (9.0%)	14.7 (9.9%)	16.2	15.2 (5.9%)	15.1 (6.6%)	14.9 (7.7%)
	15	5.7	5.4 (6.4%)	5.1 (11.5%)	5.1 (11.6%)	5.7	5.4 (5.6%)	5.1 (11.0%)	5.0 (12.4%)	5.9	5.4 (8.9%)	5.2 (11.1%)	5.0 (15.8%)
	20	2.2	2.0 (9.7%)	1.8 (16.5%)	1.9 (15.2%)	2.3	2.0 (13.8%)	1.9 (17.7%)	1.9 (16.6%)	2.3	2.0 (13.3%)	1.9 (17.1%)	1.8 (19.0%)
	25	0.2	0.1 (40.9%)	0.1 (38.0%)	0.2 (32.2%)	0.2	0.1 (40.7%)	0.1 (35.8%)	0.2 (28.0%)	0.2	0.1 (39.8%)	0.1 (34.1%)	0.1 (59.1%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
145	1	2114.7	2114.7 (0.0%)	2114.7 (0.0%)	2086.9 (1.3%)	2114.7	2114.7 (0.0%)	2114.7 (0.0%)	2086.9 (1.3%)	2114.7	2114.7 (0.0%)	2114.7 (0.0%)	2086.9 (1.3%)
	5	76.0	73.2 (3.7%)	72.8 (4.3%)	70.7 (7.0%)	76.4	73.2 (4.2%)	73.0 (4.5%)	70.7 (7.5%)	76.5	73.2 (4.3%)	72.9 (4.6%)	70.8 (7.5%)
	10	11.2	10.6 (5.3%)	10.1 (10.2%)	10.0 (10.7%)	11.3	10.6 (6.1%)	10.1 (10.8%)	10.2 (10.5%)	11.2	10.6 (4.9%)	10.5 (6.6%)	10.2 (9.0%)
	15	3.0	2.7 (8.5%)	2.6 (13.9%)	2.5 (16.0%)	3.0	2.7 (9.8%)	2.5 (16.4%)	2.6 (16.1%)	3.1	2.7 (12.5%)	2.7 (12.6%)	2.6 (16.5%)
	20	0.6	0.5 (17.4%)	0.5 (22.9%)	0.4 (34.3%)	0.6	0.5 (4.2%)	0.5 (15.5%)	0.4 (35.5%)	0.5	0.5 (0.6%)	0.5 (6.5%)	0.4 (19.4%)
	25	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)

The percent improvement is given in parenthesis. All differences in mean are statistically significant (using t-tests) when the percent improvement is non-zero.

Table 6 Inner Wait Standard Deviation (mins) for the Lightest Load Routing Policies with and without Hawkes Projections.

		$\delta = 0.5$ Minutes				$\delta = 5$ Minutes				$\delta = \infty$ (End of Conv.)			
$\mathcal{S}$	κ	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP
125	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	18.0	17.7 (1.4%)	17.5 (2.4%)	17.7 (1.8%)	18.0	17.7 (1.3%)	17.7 (1.5%)	17.6 (1.8%)	18.0	17.7 (1.4%)	17.7 (1.7%)	17.7 (1.7%)
	10	27.3	26.4 (3.2%)	26.5 (2.8%)	26.5 (3.1%)	27.6	26.4 (4.1%)	26.4 (4.2%)	26.5 (3.7%)	27.6	26.4 (4.3%)	26.9 (2.5%)	26.7 (3.5%)
	15	33.1	32.5 (1.8%)	32.3 (2.2%)	32.6 (1.4%)	33.2	32.5 (2.2%)	32.4 (2.3%)	32.4 (2.6%)	33.3	32.5 (2.6%)	33.0 (1.1%)	32.7 (1.9%)
	20	37.6	36.8 (2.1%)	36.5 (2.9%)	36.9 (1.7%)	37.7	36.8 (2.4%)	36.6 (3.0%)	36.7 (2.7%)	37.6	36.8 (2.1%)	37.3 (0.8%)	36.5 (2.8%)
	25	40.2	39.3 (2.3%)	38.8 (3.5%)	39.3 (2.2%)	39.9	39.3 (1.5%)	39.0 (2.2%)	39.2 (1.8%)	40.1	39.3 (2.0%)	39.7 (0.9%)	39.0 (2.5%)
	30	41.1	40.2 (2.1%)	39.8 (3.3%)	40.2 (2.3%)	41.3	40.2 (2.6%)	40.1 (2.8%)	40.1 (2.9%)	41.2	40.2 (2.3%)	40.8 (1.0%)	40.1 (2.6%)
135	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	17.8	17.6 (1.5%)	17.5 (1.7%)	17.4 (2.1%)	17.8	17.6 (1.1%)	17.5 (1.5%)	17.5 (1.7%)	17.9	17.6 (1.8%)	17.4 (2.5%)	17.5 (2.1%)
	10	26.4	25.6 (3.1%)	25.3 (4.1%)	25.5 (3.2%)	26.4	25.6 (3.1%)	25.4 (3.8%)	25.4 (3.8%)	26.4	25.6 (3.3%)	25.7 (2.7%)	25.7 (2.9%)
	15	32.2	30.9 (4.1%)	30.8 (4.2%)	30.8 (4.3%)	31.7	30.9 (2.5%)	30.7 (3.1%)	31.1 (1.8%)	31.9	30.9 (3.2%)	31.0 (2.9%)	31.0 (2.9%)
	20	34.7	34.1 (1.8%)	33.7 (2.9%)	34.0 (2.0%)	35.0	34.1 (2.7%)	33.6 (4.1%)	34.2 (2.5%)	35.1	34.1 (2.9%)	33.9 (3.4%)	34.1 (3.0%)
	25	36.3	35.4 (2.5%)	35 (3.6%)	35.2 (2.9%)	36.1	35.4 (1.9%)	34.8 (3.6%)	35.4 (1.8%)	36.2	35.4 (2.3%)	35.0 (3.2%)	35.2 (2.8%)
	30	36.0	35.4 (1.7%)	35 (2.9%)	35.2 (2.2%)	37.2	35.4 (4.7%)	34.8 (6.3%)	35.5 (4.5%)	36.2	35.4 (2.1%)	35.0 (3.3%)	35.2 (2.6%)
145	1	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	5	17.5	17.1 (2.3%)	17.2 (2.2%)	17.2 (2.1%)	17.7	17.1 (2.9%)	17.2 (2.5%)	17.1 (2.9%)	17.8	17.1 (3.7%)	17.1 (4%)	17.2 (3.7%)
	10	25.7	24.8 (3.6%)	24.3 (5.6%)	24.3 (5.3%)	25.6	24.8 (3.3%)	24.4 (4.8%)	24.5 (4.2%)	25.4	24.8 (2.7%)	24.4 (4.1%)	24.5 (3.8%)
	15	30.2	29.1 (3.6%)	28.7 (5.1%)	28.8 (4.8%)	30.1	29.1 (3.3%)	29.0 (3.9%)	28.7 (4.7%)	30.1	29.1 (3.1%)	28.7 (4.6%)	28.6 (5.0%)
	20	32.1	31.0 (3.4%)	30.7 (4.4%)	30.3 (5.5%)	32.0	31.0 (3.2%)	30.9 (3.5%)	30.6 (4.4%)	32.1	31.0 (3.5%)	30.6 (4.8%)	30.5 (5.0%)
	25	32.0	31.2 (2.4%)	30.9 (3.4%)	30.5 (4.8%)	32.1	31.2 (2.7%)	31.1 (3.2%)	30.9 (3.9%)	32.1	31.2 (2.8%)	30.7 (4.6%)	30.7 (4.5%)
	30	31.7	31.2 (1.6%)	30.9 (2.6%)	30.5 (4.0%)	32.3	31.2 (3.2%)	31.1 (3.7%)	30.9 (4.4%)	32.2	31.2 (3.0%)	30.7 (4.8%)	30.7 (4.7%)

The percent improvement is given in parenthesis.

Table 7 Outer Wait Standard Deviation (mins) for the Lightest Load Routing Policies with and without Hawkes Projections.

		$\delta = 0.5$ Minutes				$\delta = 5$ Minutes				$\delta = \infty$ (End of Conv.)			
$\mathcal{S}$	κ	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP
125	1	1749.8	1749.8 (0.0%)	1749.8 (0.0%)	1731 (1.1%)	1749.8	1749.8 (0.0%)	1749.8 (0.0%)	1731.0 (1.1%)	1749.8	1749.8 (0.0%)	1749.8 (0.0%)	1731.0 (1.1%)
	5	140.5	138.0 (1.8%)	139.0 (1.1%)	135.8 (3.4%)	140.4	138 (1.7%)	138.4 (1.4%)	135.3 (3.7%)	140.1	138 (1.5%)	138.2 (1.4%)	136.3 (2.7%)
	10	35.3	34.9 (1.2%)	34.2 (3.1%)	33.0 (6.6%)	36.0	34.9 (3.0%)	34.8 (3.4%)	33.8 (6.2%)	35.4	34.9 (1.6%)	34.5 (2.7%)	33.7 (4.9%)
	15	17.9	17.1 (4.3%)	17.1 (4.5%)	16.9 (5.6%)	17.7	17.1 (3.2%)	17.4 (1.5%)	16.8 (5.1%)	17.9	17.1 (4.3%)	17.4 (2.8%)	16.8 (6.3%)
	20	7.8	7.1 (9.0%)	7.0 (10.0%)	7.0 (10.7%)	8.0	7.1 (10.9%)	7.6 (4.8%)	7.1 (10.9%)	7.8	7.1 (8.5%)	7.5 (3.3%)	7.1 (8.8%)
	25	3.1	3.0 (4.8%)	2.7 (11.3%)	2.7 (12.4%)	3.2	3.0 (7.2%)	3.1 (3.5%)	2.7 (14.8%)	3.1	3.0 (6.2%)	3.1 (1%)	2.8 (10.2%)
	30	0.3	0.2 (25.4%)	0.3 (9.2%)	0.2 (48.2%)	0.2	0.2 (6.1%)	0.3 (-35.9%)	0.2 (-1.7%)	0.3	0.2 (30.8%)	0.2 (34.1%)	0.2 (31.3%)
135	1	1589.0	1589.0 (0.0%)	1589.0 (0.0%)	1571.4 (1.1%)	1589.0	1589.0 (0.0%)	1589.0 (0.0%)	1571.4 (1.1%)	1589.0	1589.0 (0.0%)	1589.0 (0.0%)	1571.4 (1.1%)
	5	113.5	112.3 (1.1%)	112.0 (1.3%)	110.0 (3.1%)	114.3	112.3 (1.8%)	112.2 (1.9%)	110.2 (3.7%)	114.0	112.3 (1.4%)	112.0 (1.7%)	110.2 (3.3%)
	10	23.7	23 (2.8%)	22.7 (4.3%)	22.9 (3.3%)	24.1	23 (4.4%)	22.8 (5.2%)	22.6 (6.1%)	23.9	23.0 (3.5%)	23.1 (3.1%)	22.8 (4.3%)
	15	9.8	9.4 (4.3%)	9.1 (7.0%)	9.0 (7.6%)	9.7	9.4 (3.3%)	9.0 (6.6%)	9.0 (7.0%)	10.0	9.4 (6.2%)	9.3 (6.4%)	8.9 (10.9%)
	20	4.1	3.8 (6.9%)	3.6 (11.9%)	3.6 (10.4%)	4.2	3.8 (10.9%)	3.7 (13.5%)	3.8 (11.3%)	4.2	3.8 (11%)	3.7 (13%)	3.6 (14.1%)
	25	0.4	0.3 (38.7%)	0.3 (34.2%)	0.3 (27.9%)	0.4	0.3 (38.8%)	0.3 (32.4%)	0.3 (23.4%)	0.4	0.3 (37.9%)	0.3 (30.8%)	0.2 (56.7%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
145	1	1450.6	1450.6 (0.0%)	1450.6 (0.0%)	1434.2 (1.1%)	1450.6	1450.6 (0.0%)	1450.6 (0.0%)	1434.2 (1.1%)	1450.6	1450.6 (0.0%)	1450.6 (0.0%)	1434.2 (1.1%)
	5	92.9	90.6 (2.5%)	90.2 (2.9%)	88.0 (5.3%)	92.7	90.6 (2.3%)	90.2 (2.6%)	88.2 (4.8%)	93.2	90.6 (2.9%)	90.6 (2.9%)	87.8 (5.8%)
	10	17.9	17.4 (2.5%)	16.5 (7.5%)	16.7 (6.6%)	17.9	17.4 (3.0%)	16.8 (6.5%)	16.8 (6.4%)	17.6	17.4 (1.4%)	17.1 (3.0%)	16.7 (5.1%)
	15	5.7	5.4 (6.2%)	5.1 (10.6%)	5.1 (11.8%)	5.7	5.4 (6.4%)	5.1 (10.8%)	5.1 (11.6%)	5.9	5.4 (8.4%)	5.4 (8.4%)	5.2 (12.1%)
	20	1.4	1.2 (15.6%)	1.1 (19.5%)	0.9 (31.3%)	1.2	1.2 (1.4%)	1.1 (9.6%)	1.2 (-5.3%)	1.1	1.2 (-2.2%)	1.1 (3.3%)	0.9 (16%)
	25	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)

The percent improvement is given in parenthesis.

Table 8 Probability of Outer Wait for the Lightest Load Routing Policies with and without Hawkes Projections.

		$\delta = 0.5$ Minutes				$\delta = 5$ Minutes				$\delta = \infty$ (End of Conv.)			
$\mathcal{S}$	κ	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP	LL	UHP	BHP	SysBHP
125	1	0.983	0.983 (0.0%)	0.983 (0.0%)	0.982 (0.0%)	0.983	0.983 (0.0%)	0.983 (0.0%)	0.982 (0.0%)	0.983	0.983 (0.0%)	0.983 (0.0%)	0.982 (0.0%)
	5	0.677	0.671 (1.0%)	0.670 (1.1%)	0.665 (1.9%)	0.678	0.671 (1.2%)	0.668 (1.5%)	0.665 (2.0%)	0.678	0.671 (1.1%)	0.669 (1.3%)	0.665 (1.9%)
	10	0.349	0.346 (0.6%)	0.345 (1.0%)	0.330 (5.2%)	0.353	0.346 (1.8%)	0.344 (2.3%)	0.337 (4.6%)	0.347	0.346 (0.2%)	0.339 (2.2%)	0.336 (3.2%)
	15	0.227	0.221 (2.6%)	0.220 (3.0%)	0.218 (3.9%)	0.225	0.221 (1.6%)	0.223 (0.6%)	0.218 (2.9%)	0.224	0.221 (1.1%)	0.222 (0.7%)	0.216 (3.3%)
	20	0.116	0.109 (6.2%)	0.110 (5.2%)	0.107 (8.1%)	0.118	0.109 (7.9%)	0.116 (2.2%)	0.110 (7.0%)	0.114	0.109 (4.4%)	0.112 (1.6%)	0.109 (4.2%)
	25	0.050	0.050 (0.2%)	0.048 (4.2%)	0.046 (9.0%)	0.052	0.050 (3.8%)	0.053 (-0.9%)	0.046 (11.8%)	0.052	0.050 (2.7%)	0.051 (0.7%)	0.048 (7.1%)
	30	0.005	0.004 (21.8%)	0.005 (2.2%)	0.003 (46.1%)	0.004	0.004 (0.3%)	0.006 (-46.9%)	0.004 (-7.9%)	0.005	0.004 (27.5%)	0.004 (33.6%)	0.004 (27.9%)
135	1	0.981	0.981 (0.0%)	0.981 (0.0%)	0.981 (0.0%)	0.981	0.981 (0.0%)	0.981 (0.0%)	0.981 (0.0%)	0.981	0.981 (0.0%)	0.981 (0.0%)	0.981 (0.0%)
	5	0.640	0.633 (1.0%)	0.632 (1.2%)	0.628 (1.8%)	0.642	0.633 (1.3%)	0.633 (1.4%)	0.629 (2.1%)	0.640	0.633 (1.0%)	0.633 (1.1%)	0.630 (1.6%)
	10	0.289	0.288 (0.6%)	0.285 (1.7%)	0.286 (1.2%)	0.292	0.288 (1.5%)	0.284 (2.8%)	0.284 (2.8%)	0.289	0.288 (0.4%)	0.287 (0.7%)	0.284 (1.6%)
	15	0.153	0.154 (-0.5%)	0.150 (1.9%)	0.150 (2.2%)	0.153	0.154 (-0.3%)	0.149 (3.0%)	0.145 (5.2%)	0.155	0.154 (1.0%)	0.152 (2.0%)	0.145 (6.8%)
	20	0.072	0.071 (2.4%)	0.068 (6.2%)	0.068 (6.0%)	0.075	0.071 (5.9%)	0.070 (7.2%)	0.069 (7.5%)	0.075	0.071 (5.6%)	0.070 (6.7%)	0.068 (9.3%)
	25	0.008	0.005 (35.7%)	0.006 (29.8%)	0.006 (24.5%)	0.008	0.005 (35.5%)	0.006 (28.3%)	0.007 (19.9%)	0.008	0.005 (34.8%)	0.006 (26.6%)	0.004 (54.7%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
145	1	0.979	0.979 (0.0%)	0.979 (0.0%)	0.979 (0.0%)	0.979	0.979 (0.0%)	0.979 (0.0%)	0.979 (0.0%)	0.979	0.979 (0.0%)	0.979 (0.0%)	0.979 (0.0%)
	5	0.604	0.595 (1.5%)	0.592 (2.0%)	0.587 (2.9%)	0.609	0.595 (2.2%)	0.594 (2.4%)	0.586 (3.7%)	0.604	0.595 (1.6%)	0.593 (1.8%)	0.588 (2.7%)
	10	0.266	0.265 (0.5%)	0.261 (1.9%)	0.262 (1.8%)	0.267	0.265 (0.6%)	0.263 (1.6%)	0.262 (1.6%)	0.267	0.265 (0.7%)	0.265 (0.6%)	0.262 (1.9%)
	15	0.109	0.105 (3.5%)	0.103 (5.1%)	0.102 (5.8%)	0.108	0.105 (3.0%)	0.103 (4.6%)	0.103 (5.1%)	0.111	0.105 (5.0%)	0.108 (2.5%)	0.104 (5.5%)
	20	0.027	0.024 (9.9%)	0.024 (11.8%)	0.021 (23.6%)	0.023	0.024 (-4.6%)	0.023 (2.1%)	0.027 (-15.1%)	0.023	0.024 (-7.4%)	0.024 (-5.6%)	0.021 (9.1%)
	25	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)
	30	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)	0	0 (0%)	0 (0%)	0 (0%)

The percent improvement is given in parenthesis.