http://pubsonline.informs.org/journal/ijoc/

INFORMS JOURNAL ON COMPUTING
Vol. ■■, No. ■■, ■■ 2018, pp. 1–17
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Virtual Statistics in Simulation via *k* Nearest Neighbors

Yujing Lin,[a] Barry L. Nelson,[a] Linda Pei[a]

[a] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208-3119

Contact: yujinglin2013@u.northwestern.edu (YL); nelsonb@northwestern.edu, http://orcid.org/0000-0002-1325-2624 (BLN); LindaPei2016@u.northwestern.edu (LP)

Received: September 21, 2017
Revised: January 30, 2018; May 23, 2018
Accepted: May 24, 2018
Published Online: ■■ ■■, 2018

https://doi.org/10.1287/ijoc.2018.0839

Abstract. "Virtual statistics," as we define them, are estimators of performance measures that are conditional on the occurrence of an event; virtual waiting time of a customer arriving to a queue at time τ_0 is one example of virtual performance. In this paper, we describe a *k*-nearest-neighbor method for estimating virtual performance postsimulation from the retained sample paths, examining both its small-sample and asymptotic properties and providing two approaches for measuring the error of the *k*-nearest-neighbor estimator. We implement leave-one-replication-out cross-validation for tuning a single parameter *k* to use for any time (or times) of interest and evaluate the prediction performance of the *k*-nearest-neighbor estimator via controlled studies. As a by-product, this paper motivates a different way of thinking about how to process the output from dynamic, discrete-event simulation.

History: Accepted by Bruno Tuffin, Area Editor for Simulation.
Funding: This research was partially supported by the National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation [Grant 1537060] and Grant Opportunities for Academic Liaison with Industry cosponsor SAS Institute.
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2018.0839.

Keywords: Simulation, Statistical Analysis • Statistics • Queues: Nonstationary

1. Introduction

The design and analysis of discrete-event, stochastic simulation has been greatly influenced by its heritage in queueing theory and the limitations of early computers. As with queueing, both the research and practice of simulation output analysis has emphasized *long-run* (or summary) performance measures for stationary service systems; these are obtained from a simulation experiment designed to precisely estimate predetermined long-run performance measures to satisfy a specific system design or improvement objective. Although some high-level characteristics of simulated systems can be represented by long-run performance measures, the underlying dynamics of the system can be masked. Also, long-run performance measures are typically not appropriate if the system of interest is inherently nonstationary.

When simulation languages were initially developed, both dynamic and persistent memory, as well as processing power, were dear, making it essential to compute and compactly report performance statistics "on the fly." This view was very compatible with a focus on predetermined long-run-average performance, but now it is outdated and limiting. As we illustrate later, retaining a complete record of every event and state change that occurs in many simulations creates what could, at most, be called "moderate-sized data" relative to current "big data" standards.

By contrast, this paper focuses on estimating a class of *time-dependent* performance measures for a (possibly) nonstationary stochastic process. The class of measures of interest we call *virtual performance* at time τ_0, denoted by $V(\tau_0)$. Specifically, $V(\tau_0)$ is some aspect of system performance *conditional* on a particular event occurring at time τ_0, where the time τ_0 is fixed and independent of the system state; it may be specified arbitrarily or perhaps be for a collection of τ_0's in a range $T_{\text{start}} \le \tau_0 \le T_{\text{end}}$. Let F_{τ_0} represent the distribution of $V(\tau_0)$. A virtual performance measure is some property of $V(\tau_0)$, such as its mean $v(\tau_0) = \mathrm{E}[V(\tau_0)]$, its q-quantile $F^{-1}_{\tau_0}(q)$, or its entire distribution F_{τ_0}. In this paper, we focus on the mean. Notice that a probability $F_{\tau_0}(x) = \Pr\{V(\tau_0) \le x\}$ can be represented as a mean.

A familiar example of $V(\tau_0)$ is the virtual waiting time of a customer arriving to a service system at time τ_0. Actually, this example is less "familiar" than one might think as there are different ways to define it. By "virtual waiting time" at τ_0, one could mean the following: (1) injected: the waiting time of a customer artificially injected into the nominal stochastic mechanism governing arrivals; (2) phantom: the observed work ahead of a phantom arrival at τ_0 that does not actually join the system; or (3) conditional: the waiting time encountered by one of the nominal stochastic arrivals, conditional on such an arrival having occurred at time τ_0. In some settings, injected and phantom are the same but

not always (e.g., queue disciples that are not first in, first out or time to traverse a network in which there is overtaking, meaning customers may depart in a different order than they arrive). Our interest is in conditional, which we define precisely in Section 3. The key distinctions are these: injected and phantom do not refer to naturally occurring arrivals, and conditional does; injected and conditional imply a customer actually arriving to the system, and phantom does not; and injected changes the queueing process by inserting an artificial customer, and conditional and phantom do not. See Section 2 for an elaboration of the distinctions.

Smith and Nelson (2015) consider the case of a traveler who wants to know how long it might take to clear security if arriving to the airport at, say, 7:30 a.m. They use simulation to answer the question by estimating the average time to clear security for passengers from the nominal airport arrival process in a one-hour time bucket containing 7:30 a.m. (i.e., no arrival is artificially injected at 7:30 a.m.). This is a conditional performance measure being used to answer an injected question, and the conditional performance is approximated by averaging over a time bucket.

A second example is the response time to a serious fire that occurs at time τ_0 as described in Carter and Ignall (1975). In their situation, the event of interest is extremely rare at time τ_0 or any other time, so the ability to *compute conditional performance measures* at any time τ_0 is critical. This is an example of phantom performance.

In the airport arrival situation considered by Smith and Nelson (2015), one expects that arrivals near 7:30 a.m. do regularly occur, and therefore, the observed waiting times of those travelers can be used to estimate properties of $V(7{:}30)$. For the rare-event situation of Carter and Ignall (1975), the method we introduce is asymptotically valid for a conditional version of the problem, but for realistic numbers of replications, few if any of the observed serious fires will be "near" any specific time τ_0 and a k-nearest-neighbor ($k$nn) estimator, such as we propose, will be significantly biased. Loosely speaking, we address situations in which the conditioning event is likely to occur near the time of interest on every simulation replication.

Estimating properties of $V(\tau_0)$ is greatly facilitated by *retaining sample path information* from many simulation replications. This is in contrast to a simulation experiment specifically designed to estimate virtual performance or virtual statistics that are computed on the fly as the simulation executes as in Smith and Nelson (2015). Although we do not introduce a specific data structure in this paper, what we have in mind is something like a time-stamped trace of all events and state changes throughout the simulation run; this sort of trace is available in nearly all commercial simulation languages although the tools to query it are not (yet). However, our methods only require a small subset of this data to be retained, as described later, so a complete trace is not necessary. Retaining, rather than summarizing, data allows us to estimate a globally optimal tuning parameter $k^\star$ for the entire time range $[T_{\text{start}}, T_{\text{end}}]$ and to quickly compute the $k$nn virtual statistic for any τ_0 in this range, whether chosen in advance or required later.

The reuse of retained simulation output data has been considered by others, notably Zhao et al. (2006, 2007), Zhao and Vakili (2008), Rosenbaum and Staum (2015), and Feng and Staum (2015). However, in these papers, the reasons to retain the data are to improve the precision of new simulation experiments (e.g., by generating control variates) or to massage the retained data so that they represent a system not actually simulated (thereby avoiding the additional simulation). Our approach is an example of "simulation analytics" (Nelson 2016). Simulation analytics treats stochastic simulation as data analytics for systems that do not yet exist and extends traditional performance estimation and system optimization to uncovering underlying patterns and the key drivers and dynamics of system behavior. See Nelson (2016) for an argument in favor of a data-analytics approach to simulation output analysis. Our virtual statistics illustrate one of the benefits of retaining simulation sample paths rather than automatically summarizing them: the ability to estimate dynamic, conditional performance and to answer questions not originally anticipated without rerunning the simulation.

The remainder of this paper is organized as follows. We start with issues in defining and estimating virtual performance in Section 2. A more detailed problem description of virtual statistics is provided, and the $k$nn point estimator for the mean of virtual performance is formally defined in Section 3. In Section 4, we present the asymptotic properties of the proposed point estimator under specific conditions on the system of interest and the growth rate of the tuning parameter k. To measure the error of the $k$nn mean estimator, we propose two variance estimators and analyze their properties in Section 5. We discuss how to adapt cross-validation (CV) to our problem setting in Section 6. To evaluate the prediction performance of the $k$nn estimator, we apply our method to controlled studies in Section 7, comparing our point estimator and a natural competitor with the true mean of virtual performance and examining the two variance estimators. Some conclusions and future work are offered in Section 8. A less fully developed and technically incomplete preliminary version of this work was reported in Lin and Nelson (2016).

2. Issues in Defining and Estimating Virtual Performance

There has been substantial work on the concept of virtual performance, including virtual waiting time, in

the queueing literature. The famous result that "Poisson arrivals see time averages" or PASTA discussed in Wolff (1982) is of this type. Loosely translated in the queueing context, PASTA indicates that the distribution of the number of customers in the system observed by arrivals from a Poisson process is the same as the time-average number of customers in the system provided the system in no way anticipates the customer's arrival. PASTA relates to what the random arrivals see, not arrivals conditioned on a fixed time as we consider here.

For a queueing system in which time-dependent state probabilities can be represented or approximated with a finite number of differential equations, whether it be Kolmogorov equations or moment differential equations, we can compute some virtual performance measures of the system using numerical integration. See Ong and Taaffe (1989) and Nelson and Taaffe (2004) for examples. We use this method to construct test problems with known solutions in Section 7 and describe how we do it in Appendix F of the online supplement.

More generally, one can consider the *work* in the system at time τ_0, where work is defined as some measure of service pending or in process; see Wolff (1989), chapters 5 and 10, for a thorough discussion. One relevant measure of work at time τ_0 is the sum of the service times of all customers in the queue plus the remaining service times of those customers in service at τ_0. In a single channel queue with no overtaking, the virtual work and the virtual (phantom or injected) waiting time at time τ_0 coincide. Just like the performance measure of number in the system, virtual work is observable at any time τ_0, so each simulated replication provides one unbiased observation.

Many real-world systems are much more complex so that PASTA or virtual work cannot be applied directly. In fact, virtual waiting time in queueing theory is typically for stationary systems in steady state. Further, the customers in a multiserver, multistation system could pass each other, so the aforementioned measure of work at time τ_0 might not coincide with the virtual waiting time at τ_0. The arrivals of interest may not even be from outside the system; they could be arrivals to an internal queue in a network that are departures from other queues. A nonqueueing context is the virtual recovery time of a manufacturing system if a failure occurs at τ_0. Although we use conditional virtual waiting time for (external or internal) arrivals to a queue as our example throughout this paper, our goal is to develop a more general approach to estimate virtual performance for complex systems. Notice that the "virtual" aspect of virtual performance in our conditional sense reflects the fact that the particular event need not actually occur in the simulated sample paths, and in many cases, it has probability zero. This is very different from, say, the waiting time of the nth arriving customer, which is observable on every replication.

There are roles for injected, phantom, and conditional virtual performance, but they are distinctly different concepts, and using one could be misleading when interpreted as the other. As a simple illustration, consider a single-queue system with a fairly regular (low-variance) arrival process having rate one; this could be achieved, say, by independent and identically distributed (i.i.d.) Erlang interarrival times for which the Erlang distribution has a large number of phases. Now suppose we are interested in the virtual waiting time of outside arrivals at a collection of times $\tau_0 = 1, 2, 3, \ldots$. Injected would insert a customer every time unit, effectively doubling the arrival rate and increasing the waiting time; in fact, the injected system might not even be stable. Injected makes the most sense when we really want to know what would happen if we impose unexpected arrivals on a background arrival process, for instance, high-priority rush orders. Phantom would not affect the arrival rate or evolution of this system but would obtain a virtual waiting time from each sample path of the queueing simulation at, say, time $\tau_0 = 5$ even if no arrival occurred at or near time $\tau_0 = 5$. Phantom is an outside observer's perspective on the (unconditional) state of the system at a particular time. Conditional, on the other hand, refers to sample paths that actually have arrivals at time $\tau_0 = 5$; therefore, conditional is *not* the outside observer's perspective but rather is from the perspective of customers who are on sample paths that yield an arrival at, say, $\tau_0 = 5$. When it is possible to define phantom, it may be close to conditional. However, if the arrival process is nonstationary with arrivals near time $\tau_0 = 5$ being rare, sample paths with arrivals near that time will be distinctly different from the generic sample path, and thus, phantom and conditional virtual waiting time will be quite different.

3. Problem and Method

We begin with an abstract definition of virtual performance, then specialize to the simulation setting. Consider a stochastic point process and an interval that begins at time $T_{\text{start}} \equiv 0$ and ends at time $T_{\text{end}} \equiv T$, where $\mathrm{E}(T^2) < \infty$. The *random* event times are $0 < t_1 < t_2 < \cdots < t_M \le T$, where M is also random; in the simulation setting, these will typically be the times that a common type of event occurs, such as "customer arrival to station 4" or "machine failure in work center G," although that is not essential. From here on, we call all of these events "arrivals" even though they may not be. Associated with event time t_i is a random performance variable $Y_i = Y(t_i)$, where we use the latter notation to emphasize the importance of time. In the simulation setting, this might be the sojourn time for the ith customer who arrived at time t_i or the time until the system is restored after the ith failure that happened at time t_i.

For a *fixed* time $0 \le \tau_0 \le T$ and a *fixed* arrival i, define the random variable

$$V_i(\tau_0) \stackrel{\mathscr{D}}{=} (Y(t_i)|t_i = \tau_0)$$

as a random variable with the same distribution as $Y(t_i)$ conditional on the timing event being equal to τ_0. For instance, if $(t_i, Y(t_i))$ have a joint density f_{t_i,Y_i} and at τ_0 the marginal density $f_{t_i}(\tau_0) > 0$, then for any $y_0 \in \Re$

$$\Pr\{V_i(\tau_0) \le y_0\} = \int_{-\infty}^{y_0} \frac{f_{t_i,Y_i}(\tau_0, y)}{f_{t_i}(\tau_0)}\, dy.$$

The random variable $V_i(\tau_0)$ could be called the virtual performance of the ith arrival at time τ_0, which might be of use in some contexts. Instead, we are interested in $V(\tau_0)$, where

$$\Pr\{V(\tau_0) \le y_0\} = \sum_{i=1}^{\infty} \Pr\{V_i(\tau_0) \le y_0\}\, q_i(\tau_0) \qquad (1)$$

and $q_i(\tau_0) = \Pr\{t_i = \tau_0 | \text{an arrival occurs at } \tau_0\}$. In words, $V(\tau_0)$ is the performance Y for an arrival at τ_0 given that *some* arrival occurred at time τ_0. We refer to this as the virtual performance at τ_0. Notice that this is conditional performance, and the conditioning is on the natural arrival process, not an injected or a phantom arrival.

Our goal in this paper is to estimate $v(\tau_0) = \mathrm{E}[V(\tau_0)]$ and also the error in our estimate of it from n independent simulation replications. Therefore, our simulation data are $\{(t_{ij}, Y(t_{ij}));\ i = 1, 2, \ldots, M_j, j = 1, 2, \ldots, n\}$, where the subscript j denotes the jth replication, and M_j is the (possibly random) number of arrivals in the jth replication. We describe our specific assumptions about this process in the next section.

Remark 1. $\{(t_i, Y(t_i));\ i = 1, 2, \ldots,\}$ looks superficially like a *marked point process (MPP)*, but this is inconsistent with standard terminology. In our setting, $\{(t_i, Y(t_i));\ i = 1, 2, \ldots,\}$ is a process that is typically derived from, or embedded in, a more complex stochastic process. Therefore, it is more like a P-MPP, a *process jointly with an MPP* (Sigman 1995). The performance measure $Y(t_i)$ depends on the complex process as well as (possibly) the "mark."

Remark 2. One may wonder when the conditional probabilities $q_i(\tau_0)$ in Equation (1) will be nonzero. A sufficient condition is that $\{t_1, t_2, \ldots, t_m\}$ have a joint density on $(0, T]$ given $M = m$ arrivals, but even weaker conditions will also suffice.

Smith and Nelson (2015) use the observed outputs $\{Y(t_{ij}) : t_{ij} \in [t_L, t_U]\}$ to estimate $v(\tau_0)$, where $t_L \le \tau_0 \le t_U$ and $[t_L, t_U]$ is a predefined time bucket. Their primary assumption is that the outputs within a time bucket are approximately stationary. There is a bias–variance trade-off in choosing $\Delta = t_U - t_L$: large Δ reduces variance but increases bias; small Δ increases variance but may reduce bias unless the probability of an empty bucket becomes too large. A modification is to use a window $[\tau_0 - \delta, \tau_0 + \delta]$ centered at τ_0 instead. This is probably an improvement and is feasible if all of the τ_0 values of interest are known in advance. The same bias–variance trade-off still exists.

In this paper, we propose constructing a $k$nn estimator from the simulation data provided that all outputs $(Y(t_{ij}), t_{ij})$ are retained. In contrast to designing a time bucket in advance, as in Smith and Nelson (2015), our method uses the average performance of the k nearest arrivals around τ_0 to estimate $v(\tau_0)$ and "tunes" the value of k using the data to minimize mean squared error.

Denote the superposed process of all the observed arrival times by

$$\mathcal{T}_n = \{t_{ij} : i = 1, 2, \ldots, M_j, j = 1, 2, \ldots, n\}. \qquad (2)$$

The $k$nn estimator of $v(\tau_0)$, $\bar{V}(\tau_0)$, is

$$\bar{V}(\tau_0) = \frac{1}{k} \sum_{\ell=1}^{k} Y(\tau_0^{(\ell,n)}), \qquad (3)$$

where $\tau_0^{(1,n)} < \tau_0^{(2,n)} < \cdots < \tau_0^{(k,n)}$ are the *sorted* k nearest neighbors to τ_0 from the superposed process $\mathcal{T}_n$ and $Y(\tau_0^{(\ell,n)})$ is the corresponding observed output for $\ell = 1, 2, \ldots, k$. Notice that the "closeness" here is based on $|\tau_0^{(\ell,n)} - \tau_0|$ regardless of replication, and ties are broken arbitrarily.

From the perspective of data analytics, $k$nn is a nonparametric supervised learning approach that is suitable for problems with low dimension and independent observations. The dimension of a problem is determined by the number of predictors included in a $k$nn model, which affects the required computer memory, computation time, and smoothness of the regression function. Because the time when trigger events, such as customer arrivals, occur is the only predictor for virtual performance, the dimension is one, which is ideal. Nevertheless, the independence assumption is usually violated because the observed outputs and predictors are obtained from a (possibly) strongly dependent sample path within each replication. Thus, dealing with the correlation among observations is one of the most challenging issues. In fact, the observations from the same replication are dependent, but the ones obtained across distinct replications are independent, so in general, the k nearest neighbors are a mix of independent and dependent observations. An alternative strategy is to choose the single nearest observation to τ_0 from each replication and then select the k nearest neighbors from among these n nearest points, implying that all the k

nearest neighbors will be independent. However, the bias of the corresponding $k$nn estimator will be larger because these k observations are not necessarily very close to τ_0, especially when the arrivals within each replication are not that dense around τ_0. We employ this alternative strategy as the natural competitor in Section 7.

From the perspective of model complexity, the number of nearest neighbors to include, k, is the single tuning parameter in $k$nn. In most situations, the optimal $k^\star$ is not obvious because the true mean of virtual performance is unknown and the data are noisy. A common approach is to evaluate many values of k and then choose the best one based on an empirical bias–variance trade-off such as empirical mean squared error (EMSE). In typical data analytics problems, the chosen value of k is small, so a direct search, say, starting with $k = 1$, is possible. However, because the superposition of arrivals from n replications may be dense around τ_0 and the observations can be quite variable, using a large number of times $\tau_0^{(\ell,n)}$ close to τ_0 to estimate $v(\tau_0)$ may greatly reduce variance without a significant impact on bias. The airport check-in problem analyzed in Lin and Nelson (2016) is an example: the optimal $k^\star$ determined via CV is 1,980. Therefore, having good insights into how the value of $k^\star$ is affected by various features of the simulation data could be helpful for saving computational effort when searching for $k^\star$, a topic discussed in Lin and Nelson (2016) but not here.

4. Asymptotic Properties of the $k$nn Estimator

In this section, we show that the proposed $k$nn estimator of the expected virtual performance is asymptotically consistent and unbiased under mild conditions on the growth rate of k and the retained sample path information $\{Y(t_{ij}), t_{ij};\ i = 1, 2, \ldots, M_j,\ j = 1, 2, \ldots, n\}$. This provides useful assurance that a $k$nn estimator is appropriate and—although not the topic of this paper—would be helpful for designing sequential procedures that increase the number of replications until a fixed precision is achieved. However, because our focus is on best using a fixed set of retained sample paths, we return to small-sample behavior in Sections 5–7.

As mentioned in Section 3, the most critical challenge for understanding properties of the $k$nn estimator comes from the *dependence* among the k nearest neighbors. Intuitively, if the time that events of interest occur has positive density around the prediction point of interest and the number of retained sample paths is much greater than the number of nearest neighbors to choose (i.e., $n \gg k$), then it is very likely that the k nearest neighbors will come from *distinct* replications so that they behave like the k nearest neighbors from independent replications. Motivated by this intuition, we first investigate how fast k can grow as $n \to \infty$ to ensure this behavior before establishing the consistency of the $k$nn estimator.

Remark 3. In general, the k nearest neighbors out of n independent observations are not independent by virtue of being the k nearest neighbors. Throughout this section, when we refer to the k nearest neighbors as being "independent" or "asymptotically independent," we mean they behave like the k nearest neighbors drawn from independent observations.

In the remainder of this section, we first define key quantities, then lay out our assumptions, and finally present the asymptotic results.

For our formulation, the arrival time is the only "predictor" in the $k$nn model, and we are interested in predicting at a fixed time or times, denoted generically by τ_0. Let the arrival-counting process from a generic replication of the dynamic system be denoted by $\{N(t): t \geq 0\}$. For any time interval $(t - w/2, t + w/2]$ with $w > 0$, let the number of arrivals within $(t - w/2, t + w/2]$ be denoted by

$$N^w(t) = N\left(t + \frac{w}{2}\right) - N\left(t - \frac{w}{2}\right).$$

If τ_0 is very close to the endpoint zero, then $t - w/2$ might be negative so that $N(t - w/2)$ is not defined. A similar issue occurs for τ_0 that is close to T. Thus, we further define $N(t) = N(0)$ for $t \leq 0$, and $N(t) = N(T)$ for $t \geq T$.

Jointly with the point process, the simulation generates an output process $Y_1, Y_2, \ldots$, and each Y_i is uniquely associated with a random arrival time t_i. Although the t_i are realized in order, $t_1 < t_2 < \cdots$, the Y_i need not be. For instance, if $t_1 < t_2$ are the arrival times of the first and second customers, and Y_1 and Y_2 their respective sojourn times, then if customers can overtake each other it is possible that Y_2 is realized before Y_1. For this reason, we write $Y(t_i)$ rather than Y_i as mentioned earlier. If the simulation represents a realizable stochastic process with well-defined initial conditions, then the joint distribution of $(t_i, Y(t_i))$ is also well defined.

Again, let $\{\tau_0^{(1,n)}, \tau_0^{(2,n)}, \ldots, \tau_0^{(k,n)}\}$ be the k nearest neighbors to τ_0 from the superposed process $\mathcal{T}_n$ with its corresponding observed output $\{Y(\tau_0^{(1,n)}), Y(\tau_0^{(2,n)}), \ldots, Y(\tau_0^{(k,n)})\}$. Define

$$W_n^k(\tau_0) = \min\left\{w: \left|\mathcal{T}_n \cap \left(\tau_0 - \frac{w}{2}, \tau_0 + \frac{w}{2}\right]\right| = k\right\}, \qquad (4)$$

representing the *smallest symmetric* interval that contains the k nearest neighbors of τ_0 chosen from $\mathcal{T}_n$. Note that this interval might exceed $(0, T]$ if τ_0 is close to the endpoint zero or T, and there might not be k total arrivals if n is too small; we let $n \to \infty$, so we ignore the latter issue. We also show that $W_n^k(\tau_0)$ converges to zero asymptotically, so the first issue does not affect the following development.

For each replication, we assume the arrival-counting process $\{N(t): t \geq 0\}$ satisfies the following properties for all $t \in (0, T]$:

$$\begin{aligned} &\Pr\{N^w(t) \geq 1\} = \lambda_t w + o(w) \quad \text{and} \\ &\Pr\{N^w(t) \geq 2\} = o(w), \end{aligned} \tag{5}$$

where $\lambda_t > 0$ is the arrival process intensity at time t. These properties are weaker than those for a Poisson process because a Poisson process also requires independent increments; we assume Equation (5) holds from here on and do not restate it. The only assumption we make about the output performance Y is that $\mathrm{E}[Y^2(t_i)] < \infty$ for all i; we refer to this assumption as $\mathrm{E}[Y^2(t)] < \infty$ from here on.

Remark 4. Our assumption about the arrival process implies that it is regular, which is easily verifiable for external arrival processes but might require some thought for internal (e.g., queue-to-queue) arrivals. Note that if the first arrival cannot occur until after some clock time $T_L > 0$, then we simply call T_L "time zero" for the purpose of our definitions.

Later, to prove asymptotic unbiasedness of the *k*nn estimator, we assume the expected value of the response surface $v(\tau)$ is Lipschitz continuous for any $\tau_1, \tau_2 \in (0, T]$, that is, $|v(\tau_1) - v(\tau_2)| \leq L_1 \cdot |\tau_1 - \tau_2|$, where $L_1 > 0$ is a finite constant. This is a mild assumption that follows if $\mathrm{E}[V_i(\tau)]$ have bounded Lipschitz constants for all i.

In the following, we prove that, if $k/n \to 0$ as both $k, n \to \infty$, then the k nearest neighbors become independent and the corresponding *k*nn estimator is pointwise consistent and asymptotically unbiased. Even if k is fixed and only $n \to \infty$, we show that the *k*nn estimator is still asymptotically unbiased, and the k nearest neighbors are independent under mild conditions.

We first establish the conditions on k and n to ensure that $Y(\tau_0^{(1,n)}), Y(\tau_0^{(2,n)}), \ldots, Y(\tau_0^{(k,n)})$ are asymptotically independent as defined in Theorem 1. Lemma 1 is a critical result.

Lemma 1. *Suppose the system of interest satisfies* $\mathrm{E}[Y^2(t)] < \infty$ *and its arrival-counting process* $\{N(t): t \geq 0\}$ *satisfies Equation* (5). *If* $k/n \to 0$ *as* $n \to \infty$, *then* $W_n^k(\tau_0) \xrightarrow{L^2} 0$, *that is,*

$$\lim_{\substack{n\to\infty \\ k/n\to 0}} \mathrm{E}[(W_n^k(\tau_0))^2] = 0,$$

and further $W_n^k(\tau_0) \xrightarrow{\text{a.s.}} 0$.

Proof. To make the development easier, consider an alternative strategy for constructing a *k*nn estimator for $v(\tau_0)$. In contrast to choosing the k nearest neighbors from the superposed process defined in Equation (2), first select the nearest neighbor of τ_0, denoted by $\tilde{\tau}_{0,j}$, from the jth replication; that is,

$$\tilde{\tau}_{0,j} = \arg\min_{t_{ij}} |t_{ij} - \tau_0|, \quad j = 1, 2, \ldots, n.$$

Next, similar to $\mathcal{T}_n$, define

$$\widetilde{\mathcal{T}}_n = \{\tilde{\tau}_{0,1}, \tilde{\tau}_{0,2}, \ldots, \tilde{\tau}_{0,n}\}. \tag{6}$$

The alternative strategy averages the k nearest neighbors from among this one nearest neighbor $\widetilde{\mathcal{T}}_n$; we refer to this as the k-of-1nn strategy and compare our estimator to it in Section 7.

The corresponding smallest symmetric interval containing the k-of-1 nearest neighbors to τ_0 is

$$\widetilde{W}_n^k(\tau_0) = \min\left\{w: \left|\widetilde{\mathcal{T}}_n \cap \left(\tau_0 - \frac{w}{2}, \tau_0 + \frac{w}{2}\right]\right| = k\right\},$$

which is an upper bound on $W_n^k(\tau_0)$; that is, $W_n^k(\tau_0) \leq \widetilde{W}_n^k(\tau_0)$ for all sample paths. Hence, to prove $W_n^k(\tau_0) \xrightarrow{L^2} 0$ or $W_n^k(\tau_0) \xrightarrow{\text{a.s.}} 0$, it suffices to prove the corresponding result for $\widetilde{W}_n^k(\tau_0)$.

Consider an arbitrary replication j and let

$$p(w) = \Pr\left\{\{t_{ij}: i = 1, 2, \ldots, M_j\} \cap \left(\tau_0 - \frac{w}{2}, \tau_0 + \frac{w}{2}\right] \neq \emptyset\right\}.$$

Assumption (5) ensures that $w > 0$ implies $p(w) > 0$. For any $\epsilon > 0$, let the actual number of points within $(\tau_0 - \epsilon/2, \tau_0 + \epsilon/2]$ from $\widetilde{\mathcal{T}}_n$ be denoted by $K_{\epsilon,n}$. Define $k = k(n)$ as a function of n; our first goal is to establish the growth rate of $k(n)$ so that

$$\lim_{n\to\infty} \Pr\{\widetilde{W}_n^k(\tau_0) > \epsilon\} = 0. \tag{7}$$

Fix $\epsilon_0 > 0$ and let $p_0 \equiv p(\epsilon_0)$. Then

$$\begin{aligned} \lim_{n\to\infty} \Pr\{\widetilde{W}_n^k(\tau_0) > \epsilon_0\} &= \lim_{n\to\infty} \Pr\{K_{\epsilon_0,n} < k(n)\} \\ &= \lim_{n\to\infty} \sum_{\ell=0}^{k(n)-1} \binom{n}{\ell} p_0^\ell (1-p_0)^{n-\ell} \\ &= \lim_{n\to\infty} \Phi\left(\frac{k(n) - 1 - np_0}{\sqrt{np_0(1-p_0)}}\right) + O(n^{-1/2}), \end{aligned} \tag{8}$$

where $\Phi(\cdot)$ is the distribution of a standard normal random variable and the $O(n^{-1/2})$ term is a consequence of the Berry–Esseen theorem (Jacod and Protter 2003). Thus, Equation (7) is achieved if

$$\frac{k(n) - 1 - np_0}{\sqrt{np_0(1-p_0)}} \to -\infty \quad \text{as} \quad n \to \infty,$$

which is equivalent to

$$\frac{k(n)}{\sqrt{n}} - \sqrt{n}p_0 = \sqrt{n}\left(\frac{k(n)}{n} - p_0\right) \to -\infty \quad \text{as} \quad n \to \infty.$$

Therefore, we need $k(n)/\sqrt{n}$ to grow strictly slower than $\sqrt{n}$, and that means $k(n)/\sqrt{n} = o(\sqrt{n})$; that is,

$$\lim_{n\to\infty} \frac{k(n)/\sqrt{n}}{\sqrt{n}} = \lim_{n\to\infty} \frac{k(n)}{n} = 0.$$

Therefore, we can get $\lim_{n\to\infty}\Pr\{\widetilde{W}_n^k(\tau_0) > \epsilon_0\} = 0$, that is, $W_n^k(\tau_0) \xrightarrow{p} 0$, as long as $k(n)/n \to 0$ as $n \to \infty$. Because $\widetilde{W}_n^k(\tau_0) \le T$ and $\mathrm{E}(T^2) < \infty$, then we can further conclude $\widetilde{W}_n^k(\tau_0) \xrightarrow{L^2} 0$, according to theorem 17.4 in Jacod and Protter (2003). Therefore, we have $W_n^k(\tau_0) \xrightarrow{L^2} 0$ for all $\tau_0 \in [0,T]$.

To prove almost sure convergence, we show in Appendix E of the online supplement that $\sum_{n=1}^{\infty}\Pr\{\widetilde{W}_n^k(\tau_0) > \epsilon_0\} < \infty$ and then apply the Borel–Cantelli theorem (Jacod and Protter 2003). □

Lemma 2. *If $k/n \to 0$ as $n \to \infty$, then the probability that any t_{ij} is one of the $k(n)$ nearest neighbors infinitely often in n is zero.*

Proof. We prove this lemma by contradiction. The smallest symmetric interval containing the single nearest neighbor to τ_0 is denoted by $W_n^1(\tau_0) > 0$. Suppose for some $\tilde{n} > 0$ the nearest neighbor will be among the $k(n)$ nearest neighbors infinitely often for all $n > \tilde{n}$ as $n \to \infty$ (clearly t_{ij} farther away have even less chance). Thus,

$$W_n^k(\tau_0) \ge W_{\tilde{n}}^1(\tau_0) > 0 \text{ infinitely often}$$

and, therefore, does not converge to zero. This contradicts Lemma 1, which shows, for any $n > 0$, $W_n^k(\tau_0) \xrightarrow{\text{a.s.}} 0$ if $k/n \to 0$ as $n \to \infty$. □

Theorem 1. *Suppose the system of interest satisfies $\mathrm{E}[Y^2(t)] < \infty$ and its arrival-counting process $\{N(t): t \ge 0\}$ satisfies Equation (5). Let*

$$I_n^k(\tau_0) = \begin{cases} 1, & \text{if } \tau_0^{(1,n)}, \tau_0^{(2,n)}, \ldots, \tau_0^{(k,n)} \text{ are} \\ & \quad \text{from distinct replications} \\ 0, & \text{otherwise.} \end{cases}$$

If $k/n \to 0$ as $n \to \infty$, then $\Pr\{I_n^k(\tau_0) = 1\} \to 1$ for any $\tau_0 \in [0,T]$; that is, the k nearest neighbors of τ_0 are asymptotically independent.

Proof. For the sake of simplicity, let $W_n^k = W_n^k(\tau_0)$ in this proof. Instead of proving $\Pr\{I_n^k(\tau_0) = 1\} \to 1$, we show the convergence of $\Pr\{I_n^k(\tau_0) = 0\}$, that is, convergence of the probability that at least one replication contributes multiple observations to the k nearest neighbors.

We have shown that the width of the interval W_n^k will converge to zero almost surely under certain conditions on k and n, so it is very likely that it will take many replications to get new observations into the interval. Thus, we are only interested in the replications that contribute observations into $(\tau_0 - W_n^k/2, \tau_0 + W_n^k/2]$, and for such replications,

$$\begin{aligned}\Pr\{N^{W_n^k}(\tau_0) \ge 2 \mid N^{W_n^k}(\tau_0) \ge 1\} &= \frac{o(W_n^k)}{\lambda_{\tau_0} W_n^k + o(W_n^k)} \\ &= \frac{o(W_n^k)/W_n^k}{\lambda_{\tau_0} + o(W_n^k)/W_n^k} \to 0 \\ &\qquad \text{as} \quad W_n^k \to 0.\end{aligned}$$

This means that if a replication is able to contribute observations into $(\tau_0 - W_n^k/2, \tau_0 + W_n^k/2]$, then the probability that this replication contributes multiple observations converges to zero as $W_n^k \to 0$. From Lemma 2, we know that the k nearest neighbors at any fixed n will eventually be replaced by new observations as n increases. Thus, the probability that there exists at least one replication contributing multiple observations converges to zero as $W_n^k \to 0$, that is, $\Pr\{I_n^k(\tau_0) = 0\} \to 0$ as $W_n^k \to 0$. Therefore,

$$\Pr\{I_n^k(\tau_0) = 1\} = 1 - \Pr\{I_n^k(\tau_0) = 0\} \to 1 \quad \text{as} \quad W_n^k \to 0. \tag{9}$$

The critical condition required for Equation (9) to hold is that the width of interval containing the k nearest neighbors should converge to zero. According to Lemma 1, $W_n^k \xrightarrow{\text{a.s.}} 0$ if $k/n \to 0$ as $n \to \infty$. Therefore,

$$\lim_{\substack{n\to\infty \\ k/n\to 0}} \Pr\{I_n^k(\tau_0) = 1\} = 1 \tag{10}$$

implying that the k nearest neighbors to τ_0 are asymptotically independent for any $\tau_0 \in [0,T]$. □

Devroye (1981) proves that the *k*nn estimator for a regression function is *pointwise consistent* if all the observations are independent and

$$\frac{k}{n} \to 0 \quad \text{as} \quad k, n \to \infty. \tag{11}$$

Refer to Appendix A of the online supplement for a restatement of Devroye's (1981) results. Although the observations are dependent in our problem setting, we have shown in Theorem 1 that the k nearest neighbors are asymptotically independent if $k/n \to 0$ as $k, n \to \infty$, which is the same condition as Equation (11). By applying Devroye's (1981) results and Theorem 1, we can establish the asymptotic consistency for the proposed *k*nn estimator for the expected virtual performance of interest.

Theorem 2. *Suppose the system of interest satisfies $\mathrm{E}[Y^2(t)] < \infty$, and its arrival-counting process $\{N(t): t \ge 0\}$ satisfies Equation (5). If $k/n \to 0$ as $k, n \to \infty$, then the knn estimator $\bar{V}(\tau_0)$ is asymptotically consistent for $v(\tau_0)$.*

Proof. According to Devroye (1981), if all observations are independent and Equation (11) is satisfied, then $\mathrm{E}(|\bar{V}(\tau_0) - v(\tau_0)|^2) \to 0$; that is,

$$\lim_{\substack{k,n\to\infty \\ k/n\to 0}} \mathrm{E}(|\bar{V}(\tau_0) - v(\tau_0)|^2) = 0. \tag{12}$$

This means that $\bar{V}(\tau_0)$ converges to $v(\tau_0)$ in mean square, and it also implies that $\bar{V}(\tau_0) \xrightarrow{p} v(\tau_0)$. Our imposed condition is $k/n \to 0$ as $k, n \to \infty$, the same as Equation (11) required by Devroye (1981). Based on Equation (10), we can show that for every $\epsilon > 0$,

$$\begin{aligned}
&\lim_{\substack{k,n\to\infty \\ k/n\to 0}} \Pr\{|\bar{V}(\tau_0) - v(\tau_0)| > \epsilon\} \\
&= \lim_{\substack{k,n\to\infty \\ k/n\to 0}} \Pr\{|\bar{V}(\tau_0) - v(\tau_0)| > \epsilon \,|\, I_n^k(\tau_0) = 1\}\Pr\{I_n^k(\tau_0) = 1\} \\
&+ \lim_{\substack{k,n\to\infty \\ k/n\to 0}} \Pr\{|\bar{V}(\tau_0) - v(\tau_0)| > \epsilon \,|\, I_n^k(\tau_0) = 0\}\Pr\{I_n^k(\tau_0) = 0\} \\
&= \lim_{\substack{k,n\to\infty \\ k/n\to 0}} \Pr\{|\bar{V}(\tau_0) - v(\tau_0)| > \epsilon \,|\, I_n^k(\tau_0) = 1\} \cdot 1 + 0 \\
&\le \lim_{\substack{k,n\to\infty \\ k/n\to 0}} \frac{\mathrm{E}[|\bar{V}(\tau_0) - v(\tau_0)|\,|\, I_n^k(\tau_0) = 1]}{\epsilon} \\
&\quad \text{(by Markov inequality)} \\
&= 0,
\end{aligned}$$

where the last equality follows from Equation (12) because when $I_n^k(\tau_0) = 1$ the k nearest neighbors are like the k nearest neighbors from independent replications and $\epsilon > 0$ is fixed. Therefore, if $k/n \xrightarrow{p} 0$ as $k, n \to \infty$, then our proposed $k$nn estimator $\bar{V}(\tau_0) \xrightarrow{p} v(\tau_0)$ for any $\tau_0 \in (0, T]$, implying that $\bar{V}(\tau_0)$ is asymptotically consistent for $v(\tau_0)$. □

Theorem 3. *Suppose that the system of interest satisfies* $\mathrm{E}[Y^2(t)] < \infty$, *its arrival-counting process* $\{N(t) : t \ge 0\}$ *satisfies Equation* (5), *and the mean of the response surface* $v(t)$ *is Lipschitz continuous with finite Lipschitz constant* $L_1 > 0$ *for any* $t_1, t_2 \in [0, T]$. *If* $\mathrm{E}[W_n^k(\tau_0)] \to 0$ *as* $n \to \infty$, *then the knn estimator* $\bar{V}(\tau_0)$ *is asymptotically unbiased for* $v(\tau_0)$.

Proof. The bias of the $k$nn estimator defined in Equation (3) is

$$\begin{aligned}
\mathrm{E}[\bar{V}(\tau_0) - v(\tau_0)] &= \frac{1}{k}\sum_{\ell=1}^{k} \mathrm{E}[v(t^{(\ell)}) - v(\tau_0)] \\
&\le \frac{L_1}{k}\sum_{\ell=1}^{k} \mathrm{E}[|t^{(\ell)} - \tau_0|].
\end{aligned}$$

Now because the distances from all the k nearest neighbors to τ_0 must be less than or equal to $W_n^k(\tau_0)$, the bias is upper-bounded by $L_1 \cdot \mathrm{E}[W_n^k(\tau_0)]$. We can similarly provide a lower bound with $-L_1$ replacing L_1. Therefore, the $k$nn estimator $\bar{V}(\tau_0)$ is asymptotically unbiased if $\mathrm{E}[W_n^k(\tau_0)] \to 0$ as $n \to \infty$. □

Notice that the proof of asymptotic unbiasedness of $\bar{V}(\tau_0)$ does not impose specific conditions on k and n. However, if Lemma 1 holds such that $W_n^k(\tau_0) \xrightarrow{L^2} 0$, then the condition $\mathrm{E}[W_n^k(\tau_0)] \to 0$ required in Theorem 3 is satisfied simultaneously.

Remark 5. Walk (2010) also provides consistency results for $k$nn applied to dependent data when $\{(\mathbf{X}_i, Y_i), i = 1, 2, \ldots\}$ are identically distributed with either ρ-mixing or α-mixing dependence. Such conditions would be appropriate for a stationary queueing system in steady state.

5. Error Measurement for the $k$nn Estimator

In this section, we provide two variance estimators for the proposed $k$nn point estimator $\bar{V}(\tau_0)$. We show the scaled limit of one of these is the same as the scaled limit of the true variance of $\bar{V}(\tau_0)$ under mild conditions; however, we show later that the second variance estimator has significantly better small-sample performance.

Let the marginal variance of the observation at $t = \tau_0$ be denoted by $\sigma^2(\tau_0)$, that is, $\sigma^2(\tau_0) = \mathrm{Var}(V(\tau_0))$; see Appendix C of the online supplement. The variance of the $k$nn estimator $\bar{V}(\tau_0)$ is

$$\tau_{n,k}^2(\tau_0) \equiv \mathrm{Var}[\bar{V}(\tau_0)] = \mathrm{Var}\left[\frac{1}{k}\sum_{\ell=1}^{k} Y(\tau_0^{(\ell,n)})\right].$$

Lemma 3. *Suppose that the system of interest satisfies* $\mathrm{E}[Y^2(t)] < \infty$, *its arrival-counting process satisfies Equation* (5), *and the marginal variance* $\sigma^2(t)$ *is Lipschitz continuous with finite Lipschitz constant* $L_2 > 0$ *for any* $t_1, t_2 \in [0, T]$. *If* $k/n \to 0$ *as* $n \to \infty$, *then*

$$\lim_{\substack{n\to\infty \\ k/n\to 0}} k\tau_{n,k}^2(\tau_0) = \sigma^2(\tau_0).$$

The proof is provided in Appendix C of the online supplement.

Our first approach is to use the sample variance of the k nearest neighbors $\{Y(\tau_0^{(\ell,n)})\}_{\ell=1}^k$; that is,

$$\hat{\tau}_{n,k}^2(\tau_0) = \frac{s_{n,k}^2(\tau_0)}{k} = \frac{1}{k(k-1)}\sum_{\ell=1}^{k} [Y(\tau_0^{(\ell,n)}) - \bar{V}(\tau_0)]^2.$$

Theorem 4. *Suppose that the system of interest satisfies* $\mathrm{E}[Y^2(t)] < \infty$, *its arrival-counting process satisfies Equation* (5), *and the expected value response surface* $v(t)$ *and the marginal*

variance $\sigma^2(t)$ are Lipschitz continuous with finite Lipschitz constants $L_1, L_2 > 0$ for any $t_1, t_2 \in [0, T]$. If $k/n \to 0$ as $k, n \to \infty$, then

$$k\hat{\tau}^2_{n,k}(\tau_0) = k \cdot \left[\frac{s^2_{n,k}(\tau_0)}{k}\right] \xrightarrow{p} \sigma^2(\tau_0)$$

implying that $s^2_{n,k}(\tau_0)$ is asymptotically consistent for $\sigma^2(\tau_0)$.

The proof is provided in Appendix D of the online supplement.

The conditions required for proving Lemma 3 and Theorem 4 imply Theorem 1, which means that the underlying assumption for proving the consistency of the sample variance estimator is that the k nearest neighbors of τ_0 should be asymptotically independent. In fact, this independence assumption is important because a sample variance computed from dependent observations is biased for the true variance.

Of course, in a finite sample, the $\{Y(\tau_0^{(\ell,n)})\}_{\ell=1}^k$ may not be independent, and thus, $s^2_{n,k}(\tau_0)$ may be biased for $\tau^2_{n,k}(\tau_0)$. Therefore, we consider a second variance estimator for $\tau^2_{n,k}(\tau_0)$ that directly accounts for this dependence using bootstrapping.

We treat each complete replication as a sample drawn from the unknown joint distribution of the stochastic system. We draw B bootstrap samples of size n, with replacement, from the original set of n replications, then compute the $k$nn estimator for each bootstrap sample and compute the sample variance of these B $k$nn estimators. Let the $k$nn estimator obtained from the bth bootstrap sample to be denoted by $\bar{V}^*_{n,k,b}(\tau_0)$. Then the bootstrap variance estimator is given by

$$\tau^{*^2}_{n,k,B}(\tau_0) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\bar{V}^*_{n,k,b}(\tau_0) - \frac{1}{B}\sum_{l=1}^{B}\bar{V}^*_{n,k,b}(\tau_0)\right)^2.$$

Because of the strong law of large numbers, we can define $\tau^{*^2}_{n,k}(\tau_0) = \lim_{B\to\infty}\tau^{*^2}_{n,k,B}(\tau_0)$ a.s. We apply these two variance estimators to empirical examples in Section 7.

6. Practical Approach

In data analytics, it is always preferred to have training data for model selection and separate testing data for assessment to avoid overfitting. When this is not possible or desirable, CV, which is an out-of-sample technique to assess prediction error, can be used to reduce overfitting. However, training the model with such methods can be computationally expensive. In this section, we discuss how to apply CV in our problem setting.

A traditional K-fold CV approach randomly divides N observations into K subsets or "folds" of size N/K, then repeatedly uses each subset of $(K-1)$ folds of data to train the model and the remaining fold as the testing data, in which K for the number of folds should not be confused with k, the number of nearest neighbors to average in a $k$nn model. This procedure is repeated K times until all folds of data are tested; then a goodness-of-fit measure is used to compare alternative models. Tenfold CV is a common choice, and it is sometimes preferred to leave-one-observation-out CV (which could be thought of as N-fold CV) because, otherwise, the folds are highly correlated because they share $(N-1)$ observations.

We should not directly apply traditional CV to virtual statistics because of the correlation among observations collected from within the same replication. Hart (1991) points out that CV performs poorly with correlated data. Thus, instead of leaving individual observations out, we propose to leave out n/K *entire replications*, using the remaining replications as the training data and the left-out replications as the testing data; recall that replications are independent. This guarantees independence of each training and testing set.

An underlying principle of CV is that the model fit obtained when leaving out some data should be representative of the model fit we would obtain using all of the data; even tenfold CV uses 90% of the data for fitting. Notice, however, that in our context the training data consist of the superposition of all of the observations from multiple replications, so the k nearest neighbors of any test point will, typically, not come from different replications and, therefore, will be correlated. For this reason we further propose *leave-one-replication-out (LORO) CV*, which increases the likelihood that the predictive performance of a value of k based on $(n-1)$ replications will be most similar to its performance with n replications. If we were to leave out a larger number of replications, then the k nearest neighbors from the remaining replications would be more strongly correlated than they would be leaving out fewer. Stated differently, if each fold leaves out one replication for testing, then there are a large number of $(n-1)$ replications for training, and the k nearest neighbors should be spread out among them as they would be with all n replications. Traditional leave-one-observation-out CV does not have this advantage in most data-analytics problems because data are usually assumed i.i.d. The details of our approach can be found in Algorithm 1. Algorithm 1 searches for a single globally optimal $k^\star$ that minimizes the EMSE(k) across the entire time range $(0, T]$; the EMSE(k) is obtained via CV in which the values of $Y(t_{ij})$ at times t_{ij} for each left-out replication j are predicted by a trial value of k applied to the data from the $(n-1)$ other replications without j.

Algorithm 1 ($k$nn Method via LORO CV)

1: Input search range $k_L < k_U$, NN = "nearest neighbors."
2: **For** $j = 1, 2, \ldots, n$, **do**
3: $S_{\text{test}} \leftarrow \{Y(t_{ij}), t_{ij}; i = 1, 2, \ldots, M_j\}$.
4: $S_{\text{train}} \leftarrow$ all data except S_{test}.

5: Find k_U NN in S_{train} to $t_{ij} \in S_{\text{test}}$.
6: Store the indices of the k_U NN to each $t_{ij} \in S_{\text{test}}$ into an index matrix $\mathbf{M}_{\text{ind}} \in \Re^{|S_{\text{test}}| \times k_U}$, where the ith row in $\mathbf{M}_{\text{ind}}$ contains the indices of the k_U NN to $t_{ij} \in S_{\text{test}}$.
7: **For** $k \in [k_L, k_U]$**, do**
8: Extract the first k columns from $\mathbf{M}_{\text{ind}}$.
9: Find the k NN to each $t_{ij} \in S_{\text{test}}$ and compute the $k$nn estimator $\bar{V}(t_{ij}, k)$.
10: **end For**
11: **end For**
12: **For** $k \in [k_L, k_U]$**, do**
13: Compute $\text{EMSE}(k) = \left(\sum_{j=1}^{n}\sum_{i=1}^{M_j}[Y_{ij} - \bar{V}(t_{ij},k)]^2\right) / \left(\sum_{j=1}^{n} M_j\right)$.
14: **end For**
15: Choose $k^\star$ that results in the minimum $\text{EMSE}(k)$.

We know that one critical issue about LORO CV is its computational effort, especially when both k and n could be very large. Thus, using a K-fold CV might be more appropriate in terms of computational effort in some situations. Refer to Hastie et al. (2001) for more discussion on the choice between K-fold CV versus leave-one-out CV from the perspective of bias–variance trade-off and computational effort. Notice that we focus on LORO CV in the remainder of the paper because, in our experiments, the number of replications n is only 10 to 100 (not a large n), so the computational effort is not a concern.

Because we suggest CV for parameter tuning, the constructed $k$nn mean estimator is actually a *cross-validated* estimator. Hence, it is also important to understand the properties of such a $k$nn estimator. Li (1984) proves asymptotic consistency for a cross-validated $k$nn mean estimator under certain conditions on the data and the weight function of the k nearest neighbors. All of the required conditions, except that the k nearest neighbors should be independent, are satisfied in our problem setting. Refer to Appendix B of the online supplement for the verification of each condition.

Thus, to apply the results from Li (1984), we need additional conditions to ensure the independence among the k nearest neighbors first. Recall that in Theorem 1 we have shown that the k nearest neighbors will be asymptotically independent as long as $k/n \to 0$ as $n \to \infty$. We have not shown that cross-validation leads to a choice of k with this property, but if it does, then Theorem 1 implies that all of the conditions of Li (1984) are satisfied. This, in turn, would imply that our proposed cross-validated $k$nn estimator is asymptotically consistent, that is,

$$\frac{1}{\sum_{j=1}^{n} M_j}\left(\sum_{j=1}^{n}\sum_{i=1}^{M_j}[v(t_{ij}) - \bar{V}(t_{ij}, k^\star_{\text{CV}})]^2\right) \xrightarrow{p} 0,$$

where $k^\star_{\text{CV}}$ is the optimal tuning parameter obtained using Algorithm 1.

7. Experiments

In this section, we describe controlled studies based on queueing models to evaluate the performance of our proposed $k$nn estimator $\bar{V}(\tau_0)$ and the corresponding variance estimators. Although the models themselves are simple, they allow us to stress the method by varying the factors that could affect estimator performance, including severity of the nonstationarity, variability, and correlation of the output response (waiting time), density of arrivals, and number of replications. In addition, we can compute the true value of $v(\tau_0)$, as described herein, which facilitates evaluating the bias of $\bar{V}(\tau_0)$.

As mentioned in Section 2, the Kolmogorov forward equations (KFEs) of a continuous-time Markov chain can be numerically integrated to obtain state probabilities over time, and from these, the true values of some virtual performance measures, such as mean virtual waiting time, can be computed. To extend to queues with non-Markovian behavior, we employ phase-type (Ph) distributions, because Ph distributions can approximate nonexponential distributions while still allowing state probabilities to be represented by KFEs; this is at the cost of expanding the state space and number of differential equations.

For arrivals, we consider homogeneous customers with either *nonstationary, two-phase hyperexponential* ($H_2(t)$) or *nonstationary, two-phase Erlang* ($E_2(t)$) interarrival times. There are s servers with stationary, exponentially distributed service times, and a single first-come, first-served queue with finite system capacity c. We denote these two queueing systems by $H_2(t)/M/s/c$ and $E_2(t)/M/s/c$, respectively. Compared with the arrivals to an $M(t)/M/s/c$ queueing system, from the perspective of coefficient of variation (cv), a $H_2(t)/M/s/c$ system is more variable with $\text{cv}_{\text{H}} > 1$, and an $E_2(t)/M/s/c$ system is less variable with $\text{cv}_{\text{E}} < 1$, so they can represent more general systems with nonexponential behavior. We also report results from an $E_4(t)/E_4/s/c$ queue for reasons described herein.

For these phase-type queueing systems, we are interested in estimating the *mean virtual waiting time* of arrivals. Because the systems have finite capacity, the virtual arrivals occurring when the system is full cannot enter the queue. The true virtual waiting time, $v(\tau_0)$, is computed to high numerical accuracy using code we developed in Python; details about the computation of $v(\tau_0)$ are offered in Appendix F of the online supplement.

To make $H_2(t)/M/s/c$ and $E_2(t)/M/s/c$ comparable in terms of their arrival processes, we first specify the two arrival-rate functions $\lambda^{(1)}_{\text{H}}(t)$ and $\lambda^{(2)}_{\text{H}}(t)$ for $H_2(t)/M/s/c$, and then the arrival-rate function for $E_2(t)/M/s/c$ is $\lambda_{\text{E}}(t) = p\lambda^{(1)}_{\text{H}}(t) + (1-p)\lambda^{(2)}_{\text{H}}(t)$, where p is the mixing probability for the hyperexponential distribution.

We study the performance of the $k$nn estimator for the virtual waiting time $v(\tau_0)$ along with the two variance

estimators under different system characteristics. A summary of all cases is provided in Table 1. For the arrival-rate functions, we test the two patterns of time-varying arrival rates presented in Figure 1—one is piecewise constant (pw-c) with different shapes, and the other is piecewise linear (pw-l) with the same shape. Notice that multiple values are tested for some parameters in some cases. For example, the first case shown in Table 1 indicates that the pw-c arrival-rate functions (Figure 1(a)) are applied to the corresponding $H_2(t)$ and $E_2(t)$ interarrival-time distributions and that these two queueing models are tested under two different values of service rate μ (i.e., 10 and 20).

Overall, the $k$nn estimator turns out to estimate $v(\tau_0)$ well across all seven cases, and the bootstrap variance estimator performs much better than the sample variance estimator. For the purpose of illustration, we provide detailed results for cases 2 and 7. The time-varying pw-l arrival-rate functions we used for these cases are presented in Figure 1(b), from which we see that the arrivals are generated from $t = 0$ to $t = 16$. We let the simulation model run until the service for all arrivals before time 16 are completed, implying that the actual run length T is greater than 16. We run the simulation in this way because if the simulation model stopped running at $t = 16$ then we would lose the data for the arrivals occurring late, especially when systems are congested. As for the other parameters, we have service rate $\mu = 20$, number of servers $s = 1$, system capacity $c = 50$, mixing probability $p = 0.4$ for the hyperexponential cases, and $B = 2,000$ resamples for computing the bootstrap variance estimator. For case 2, we test different numbers of replications: $n = 10$, implying relatively sparse superposed arrivals, and $n = 100$, yielding much denser superposed arrivals. For case 7, we only report results for $n = 10$.

We begin with the case 2 results. Figure 2 gives an illustration of the superposed sample paths from $n = 10$ replications of the $H_2(t)$ and $E_2(t)$ systems, from which we find that the $H_2(t)/M/1/c$ waiting times are more variable than $E_2(t)/M/1/c$ as expected. To assess variance-estimator performance, we ran $R = 100$ macroreplications for all the scenarios so that we could obtain an unbiased estimator for $\tau^2_{n,k}(\tau_0)$, the true variance of $\bar{V}(\tau_0)$, that is,

$$\hat{\tau}^2_R(\tau_0) = \frac{1}{R-1}\sum_{r=1}^{R}[\bar{V}_r(\tau_0) - \bar{\bar{V}}_R(\tau_0)]^2,$$

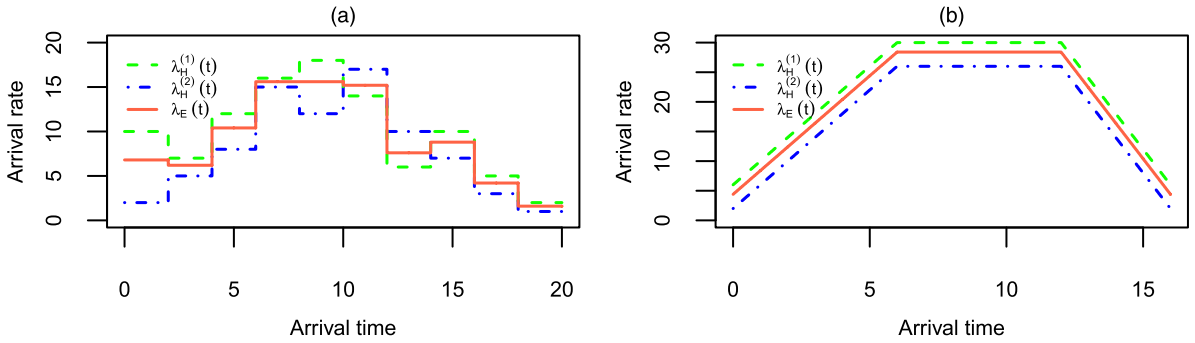
where $\bar{V}_r(\tau_0)$ is the $k$nn estimator for the expected virtual waiting time computed from the rth macroreplication and $\bar{\bar{V}}_R(\tau_0)$ is the average of these R $k$nn estimators. In each macroreplication, the optimal $k^\star$ was tuned via LORO CV using Algorithm 1. Notice that $\hat{\tau}^2_R(\tau_0)$ is available in experiments in which we make macroreplications but would not be available to the practitioner who has a single set of replications.

We estimate virtual waiting time at time points $\tau_0 = 1, 2, \ldots, 15$. Figure 3 shows the comparison between $\bar{\bar{V}}_R(\tau_0)$ and the true virtual waiting time $v(\tau_0)$. We have also added error bars at $\pm 2\hat{\tau}_R(\tau_0)/\sqrt{R}$ for

Table 1. Simulation Experiment Designs for Controlled Studies

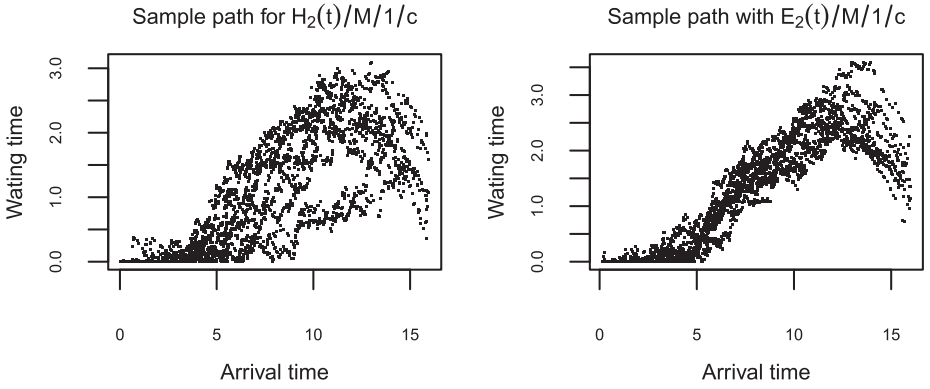
Case	Arrival rate	Arrivals	Service	μ	s	c	p	n
1	pw-c	$H_2(t), E_2(t)$	M	10, 20	1	30	0.5	50
2	pw-l	$H_2(t), E_2(t)$	M	20	1	50	0.4	10, 25, 50, 100
3	pw-l	$H_2(t), E_2(t)$	M	10	2	50	0.4	10
4	pw-l	$H_2(t), E_2(t)$	M	5	4	50	0.4	10
5	pw-l	$H_2(t), E_2(t)$	M	40	1	50	0.4	10
6	pw-l	$H_2(t), E_2(t)$	M	20	2	50	0.4	10
7	pw-l	$E_4(t)$	E_4	20	1	50	—	10

Figure 1. (Color online) Arrival Rate Functions



Piecewise-constant arrival rates. Piecewise-linear arrival rates.

Figure 2. Sample Paths with 10 Superposed Replications for Two Systems



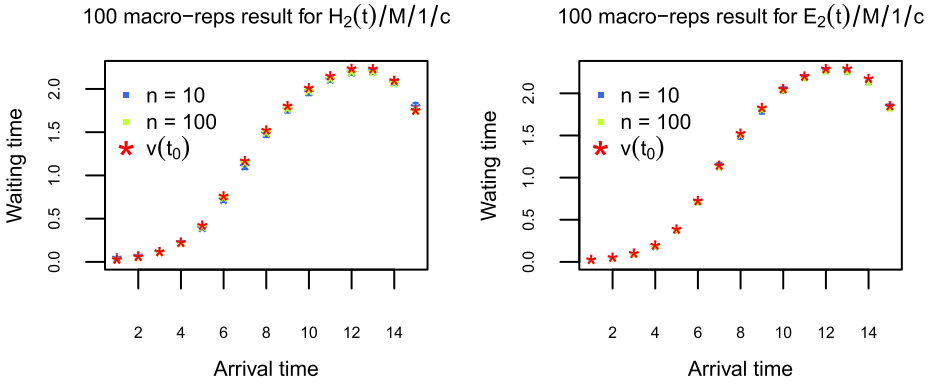
$\bar{\bar{V}}_R(\tau_0)$; however, there is so little error that they are not really visible. We see that the averaged $k$nn estimator $\bar{\bar{V}}_R(\tau_0)$ with a globally optimal $k^\star$ is effectively equal to $v(\tau_0)$ for both $H_2(t)/M/1/c$ and $E_2(t)/M/1/c$.

The result shown in Figure 3 reflects the performance of our $k$nn estimator averaged over 100 independent data sets. However, in applications, the practitioner will use a single set, so it is important to examine the estimation performance based on a single simulation. For both $H_2(t)/M/1/c$ and $E_2(t)/M/1/c$ systems, we present a good case and a bad case in terms of the performance of the $k$nn estimator for $n = 10$ and $n = 100$. For each case, we also provide a comparison between the two error estimators, $s_{n,k^\star}(\tau_0)/\sqrt{k^\star}$ and $\tau^*_{n,k^\star}(\tau_0)$, and the unbiased error estimator $\hat{\tau}_R(\tau_0)$. See Figure 4 for $H_2(t)/M/1/c$ and Figure 5 for $E_2(t)/M/1/c$.

First, consider the results for the $H_2(t)/M/1/c$ system presented in Figure 4: we see that the $k$nn estimator is close to the true value $v(\tau_0)$ for either $n = 10$ or 100 if a "good" data set is used. Even for the two "bad" cases, the $k$nn estimator is still accurate for most values of τ_0, especially when $n = 100$. As for the two error estimators, we see that the sample variance estimator, $s_{n,k^\star}(\tau_0)/\sqrt{k^\star}$, dramatically underestimates $\hat{\tau}_R(\tau_0)$, and the bootstrap variance estimator $\tau^*_{n,k^\star}$ is much closer to $\hat{\tau}_R(\tau_0)$. This is because the optimal $k^\star \gg n$ so that many of the selected observations come from the same replication; hence, the sample variance estimator underestimates the true variance because of the positive correlation among those $k^\star$ observations. As for the bootstrap variance estimator, because we bootstrap the n independent *replications*, the bootstrap variance estimator can handle the dependence more appropriately. Therefore, although we can establish asymptotic consistency for the sample variance estimator, it is less useful than the bootstrap variance estimator in practice because the data are finite and usually correlated. Further, we notice that the estimation error drops more than 100% when n increases from 10 to 100. Because the arrivals around τ_0 get denser with more replications, the $k$nn estimator $\bar{V}(\tau_0)$ is constructed from nearer neighbors; thus, $\bar{V}(\tau_0)$ is very likely to be less biased. Moreover, we find that the optimal $k^\star$ grows slower than n, as shown in Table 2, so the dependence issue among the selected $k^\star$ nearest observations should be less severe when n goes from 10 to 100. Therefore, having more replications of data reduces the error of the $k$nn estimator significantly, especially when the system of interest is variable. Similar analysis applies to the $E_2(t)/M/1/c$ system presented in Figure 5.

According to the coefficient of variation, the $E_2(t)/M/1/c$ is less variable than $H_2(t)/M/1/c$, whereas

Figure 3. (Color online) Comparison Between $\bar{\bar{V}}_R(\tau_0) \pm 2\hat{\tau}_R(\tau_0)/\sqrt{R}$ and $v(\tau_0)$ for Two Systems



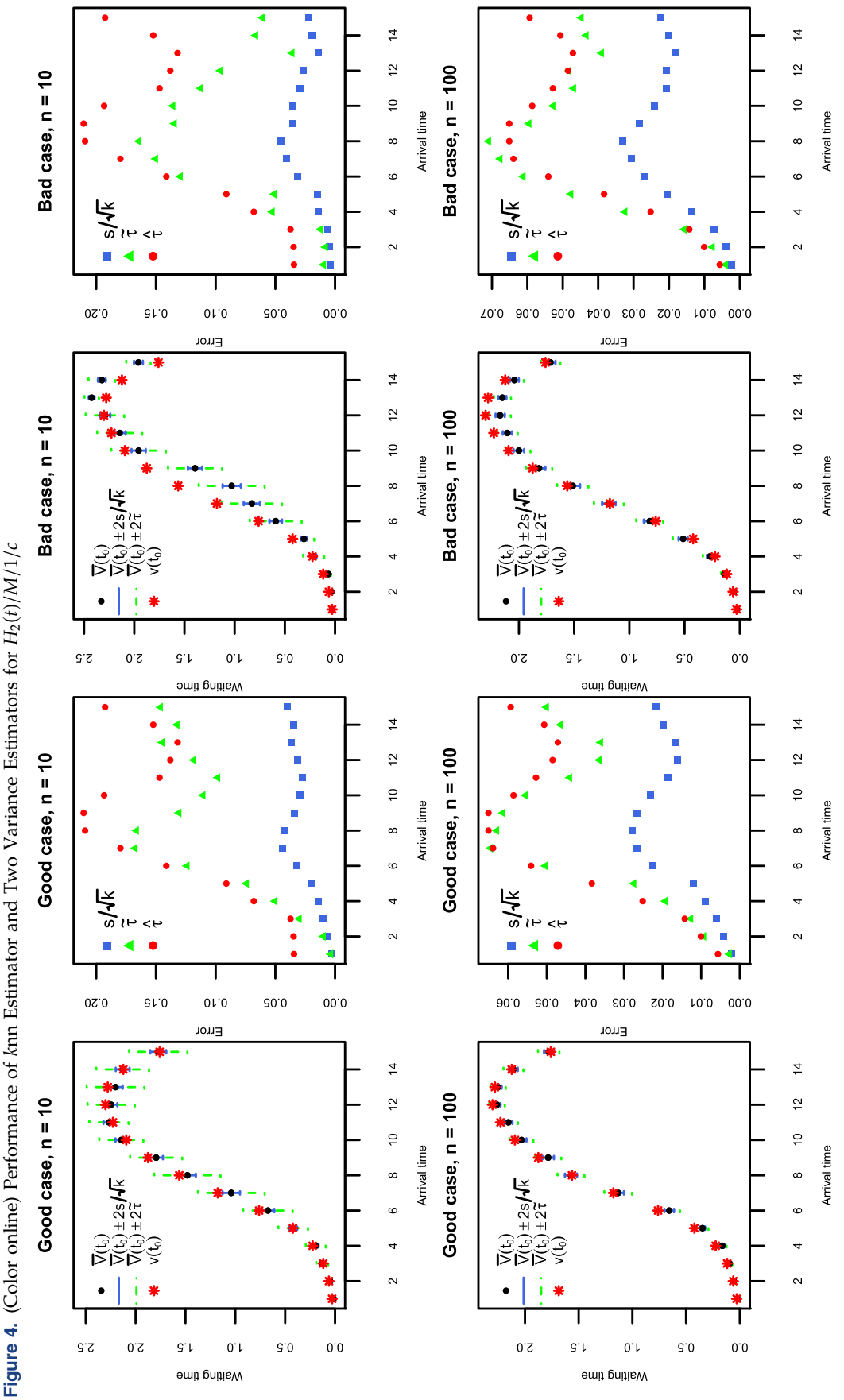


Figure 4. (Color online) Performance of $k$nn Estimator and Two Variance Estimators for $H_2(t)/M/1/c$

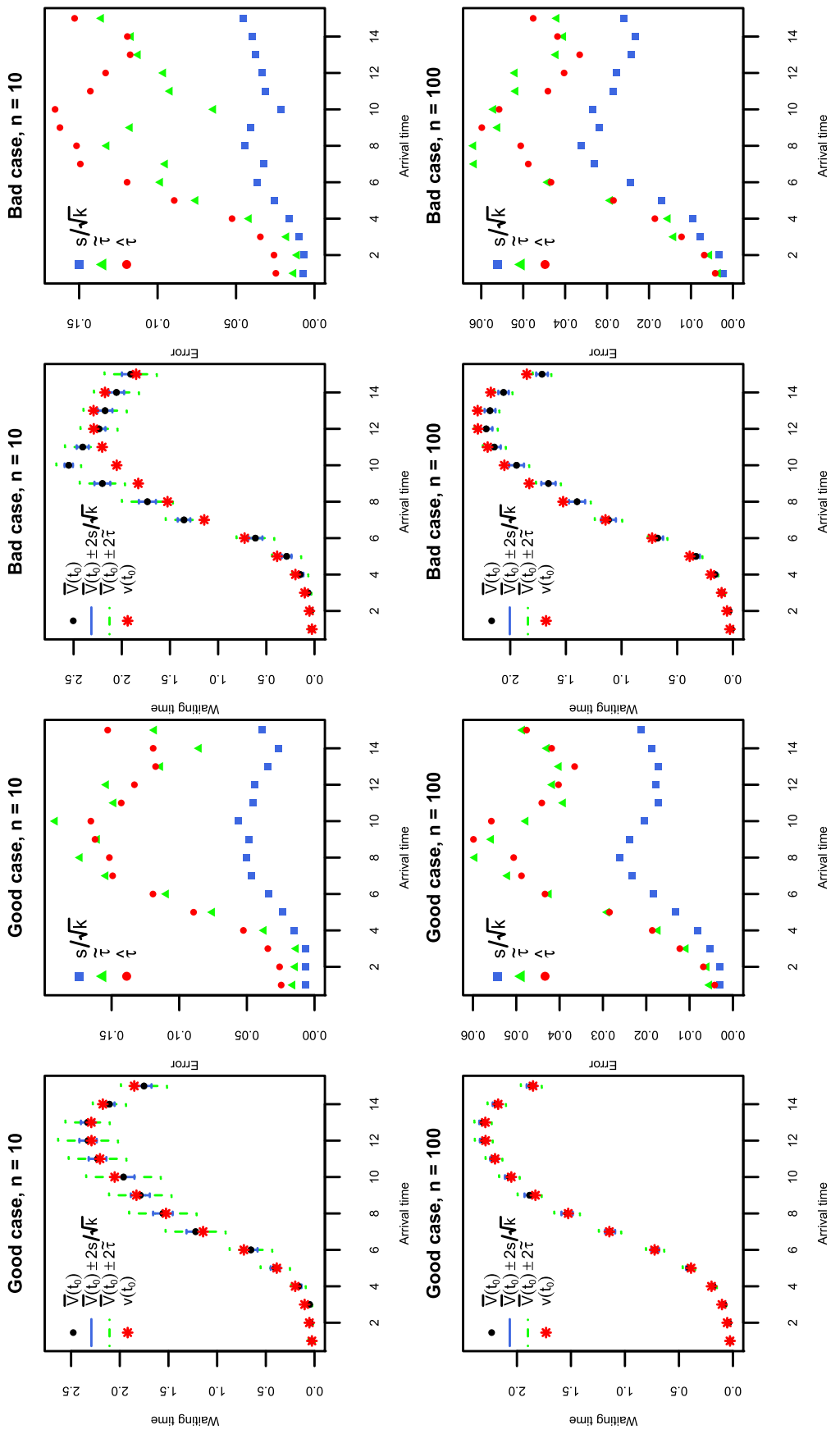


Figure 5. (Color online) Performance of $k$nn Estimator and Two Variance Estimators for $E_2(t)/M/1/c$

Table 2. Average Globally Optimal $k^\star$ for Different n

n	$H_2(t)/M/s/c$	$E_2(t)/M/s/c$
10	197	136
25	208	164
50	351	245
100	509	418

the performance of our $k$nn estimator along with its two variance estimators for $E_2(t)/M/1/c$ is not that different from the $H_2(t)/M/1/c$. Therefore, we explore an even less variable system, case 7, $E_4(t)/E_4/1/c$, which has the same system parameters but with different distributions for the interarrival time and service time. From the 100-replications-superposed sample path shown in Figure 6, we find that $E_4(t)/E_4/1/c$ is much less variable than $E_2(t)/M/1/c$. As for the $\bar{\bar{V}}_R(\tau_0)$ with a globally optimal $k^\star$ averaged over 100 macroreplications, it is also very close to the true virtual waiting time $v(\tau_0)$.

Similar to case 2 discussed previously, we present two macroreplications with $n = 10$ to illustrate good and bad performance of the $k$nn estimator and the corresponding error estimators for this $E_4(t)/E_4/1/c$ system. Compared with $n = 10$ cases shown in Figure 5, we see that the error of $\bar{V}(\tau_0)$, especially the bootstrap error estimator, reduces significantly in Figure 7. Although the sample variance estimator still underestimates the true variance, it is less biased than for the $E_2(t)/M/1/c$ system. This is because $E_4(t)/E_4/1/c$ is much less variable than $E_2(t)/M/1/c$, implying that the optimal $k^\star$ for $E_4(t)/E_4/1/c$ should be smaller than the $k^\star$ for $E_2(t)/M/1/c$. When $n = 10$, the average $k^\star$ over 100 macroreplications for $E_4(t)/E_4/1/c$ is around 48, and the one for $E_2(t)/M/1/c$ is around 136. Thus, the dependence issue among the $k^\star$ nearest neighbors in the $E_4(t)/E_4/1/c$ system is definitely less severe, and the sample variance estimator is less biased.

Overall, we see that our proposed $k$nn estimator tuned via LORO CV estimates the true value $v(\tau_0)$ well. Although the sample variance estimator has a nice asymptotic property, the bootstrap variance estimator works much better in practice because of the dependence among data. For cases 2–4 with $s \times \mu = 20$ and cases 5 and 6 with $s \times \mu = 40$, we find that the more servers the systems have, the shorter the average virtual waiting times are. Further, the systems in cases 5 and 6 serve customers very fast, such that most waiting times are pretty short or even zero; thus, there is not much trend but high variability in the sample path. Our $k$nn estimator works well for most virtual arrivals occurring during the middle of the time period but is more biased at the two boundaries. This is because the globally optimal $k^\star$ becomes very large because of the high variability of the system and such a large $k^\star$ averages too many observations for those τ_0's close to the boundaries, making the corresponding $k$nn estimator relatively more biased. In addition to the large bias at the boundaries, the bootstrapped error bar is very wide, which also reflects the high variability of the system. Also, the bootstrap variance estimator turns out to be much larger than the sample variance estimator. Again, this is because the globally optimal $k^\star$, which is around 700, is much larger than the number of replications ($n = 10$), such that the dependence issue is very severe. Hence, the sample variance estimator becomes extremely biased, but the bootstrap variance estimator still works well. For such highly variable systems, having more replications of data greatly improves the accuracy of the $k$nn estimator. On the other hand, when the system of interest is not that variable, such as the $E_4(t)/E_4/1/c$, a relatively small number of replications can provide an accurate $k$nn estimator with small error.

How sensitive are our results to the estimated value of $k^\star$ from LORO CV? The answer is not very. In Appendix H of the online supplement we report the EMSE for case 2 simulations in which we employ $\lfloor k^\star/2 \rfloor$ and $2k^\star$ as well as $k^\star$ and show that the results are robust. Recall that we estimate a single, global $k^\star$ for all target time points, so it is comforting that the results are not highly sensitive.

Figure 6. (Color online) Sample Path and Macroreplication Results with $n = 10$ for $E_4(t)/E_4/1/c$

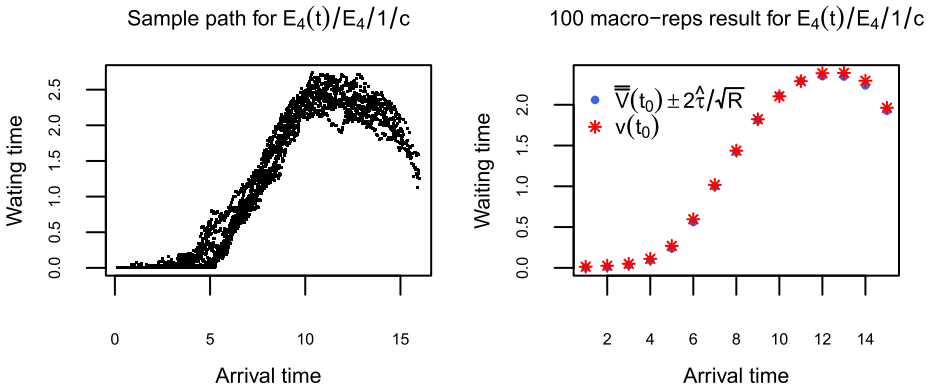
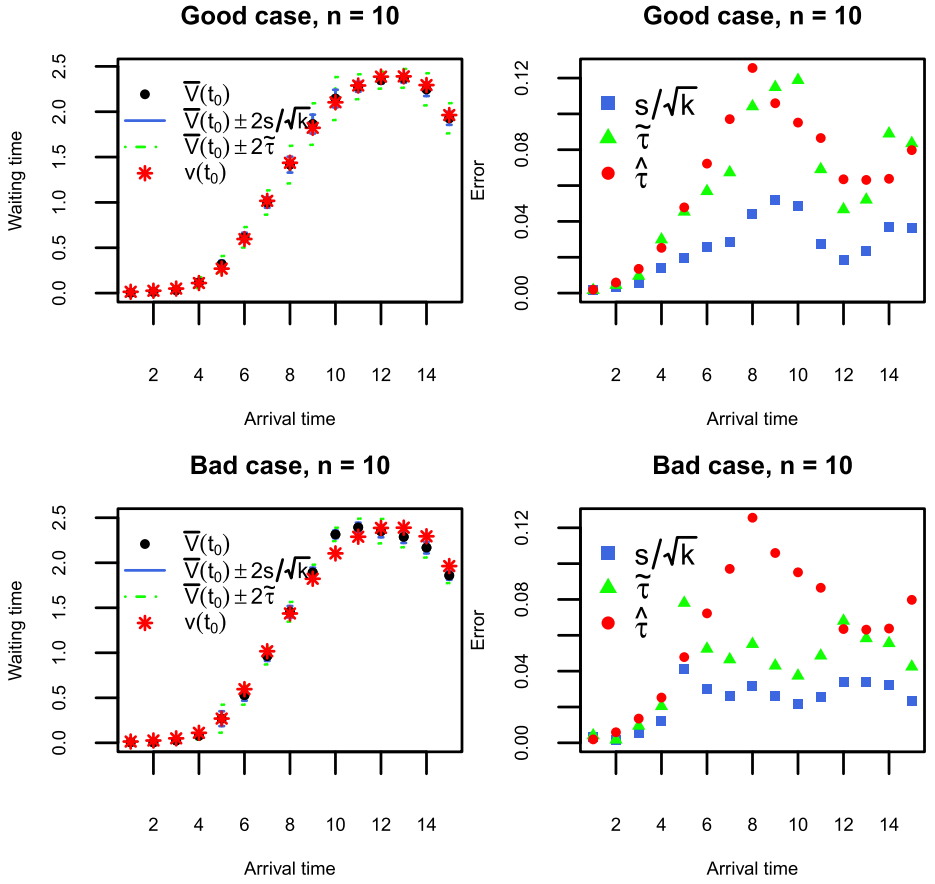


Figure 7. (Color online) Performance of *k*nn Estimator and Two Variance Estimators for $E_4(t)/E_4/1/c$



Up to this point, we have examined how properties of the queueing process affect our *k*nn estimator. Next, we compare it to the natural alternative that averages the k nearest neighbors from among the one nearest neighbor from each replication, which we call the *k*-of-1nn strategy. Because replications are independent, *k*-of-1nn is just standard k nearest neighbors for which all the usual asymptotic theory holds and we can

Table 3. $\sqrt{\text{EMSE}}$ of *k*nn and *k*-of-1nn Estimators

	$E_2(t)/M/1/c$				$H_2(t)/M/1/c$			
	$n = 10$		$n = 100$		$n = 10$		$n = 100$	
τ_0	*k*nn	*k*-of-1nn	*k*nn	*k*-of-1nn	*k*nn	*k*-of-1nn	*k*nn	*k*-of-1nn
1	0.025	0.030	0.010	0.109	0.038	0.036	0.006	0.069
2	0.026	0.054	0.012	0.128	0.035	0.063	0.010	0.098
3	0.036	0.116	0.018	0.118	0.037	0.093	0.015	0.137
4	0.057	0.186	0.026	0.177	0.068	0.182	0.026	0.180
5	0.090	0.326	0.034	0.252	0.095	0.284	0.040	0.270
6	0.120	0.410	0.049	0.352	0.147	0.447	0.057	0.392
7	0.148	0.581	0.055	0.445	0.189	0.584	0.067	0.511
8	0.155	0.579	0.056	0.486	0.212	0.628	0.069	0.587
9	0.166	0.505	0.065	0.493	0.213	0.563	0.072	0.567
10	0.165	0.467	0.060	0.456	0.198	0.538	0.067	0.495
11	0.143	0.416	0.053	0.384	0.152	0.475	0.060	0.411
12	0.134	0.386	0.054	0.343	0.144	0.415	0.062	0.368
13	0.122	0.349	0.057	0.321	0.135	0.401	0.059	0.397
14	0.126	0.382	0.065	0.309	0.153	0.444	0.058	0.394
15	0.153	0.437	0.060	0.433	0.197	0.463	0.059	0.489

directly apply leave-one-observation-out CV to tune $k^\star$; see Appendix G of the online supplement for details. Table 3 shows the EMSE of both estimators based on 100 macroreplications, for the $E_2(t)/M/1/c$ and $H_2(t)/M/1/c$ queues of case 2. Notice that *k*nn has smaller EMSE, often by as much as an order of magnitude, except for $n = 10$ replications when the target time $\tau_0 = 1$, when they are comparable. In these examples, the arrival rate is quite low near the origin, so *k*nn does benefit enough from superposing dense arrivals to be superior to *k*-of-1nn at this time point. However, overall, the benefit for choosing the *k* nearest neighbors from among *all* arrivals is substantial.

Remark 6. We mentioned the need to store detailed sample paths in Section 1. Take the $E_2(t)/M/1/c$ queueing system as an example. As an illustration, we retained a detailed sample path (i.e., a time-stamped trace of *all* events and state changes throughout one simulation run) generated by the commercial simulation software Simio, which amounted to only 5.05 MB. Thus, the total amount of data across 100 replications is about 505 MB, which is not large compared with the standards of data analytics. In addition, these sample paths generated by Simio are not stored in an efficient data structure; if they were, then the size of the data could be further reduced, and we could query the data more efficiently.

8. Conclusions

In this paper, we propose a *k*nn method for estimating virtual mean performance based on retained transactional data from simulation experiments. We derive the asymptotic properties of the *k*nn estimator and propose two variance estimators. The controlled studies show that even with a globally optimal $k^\star$, the *k*nn estimator can be very close to the true value of the virtual performance throughout the time range, and the bootstrap variance estimator performs much better than the sample variance estimator because of the dependence among data.

However, to make searching for $k^\star$ computationally feasible, we need a robust starting *k* or at least a modest range of *k* within which to search because an exhaustive search and CV calculation is $O((\sum_{j=1}^{n} M_j)^2)$. Our algorithm includes such a range, $[k_L, k_U]$, but setting this range, perhaps using results in Lin and Nelson (2016), is still an open problem. Furthermore, it will be valuable to develop an adaptive *k*nn algorithm in the sense that the optimal $k^\star$ could be tuned using a subset of data around the point of interest τ_0 instead of the entire data set.

Acknowledgments

Portions of this article were published in the *Proceedings of the 2016 Winter Simulation Conference* as Lin and Nelson (2016). The authors thank Mike Taaffe of Virginia Tech for help with creating the virtual waiting time test cases, Ohad Perry for helpful discussion about the problem definition, and the area editor, associate editor, and two referees for advice that improved the paper.

References

Carter G, Ignall EJ (1975) Virtual measures: A variance reduction technique for simulation. *Management Sci.* 21(6):607–616.

Devroye L (1981) On the almost everywhere convergence of nonparametric regression function estimates. *Ann. Statist.* 9(6): 1310–1319.

Feng M, Staum J (2015) Green simulation designs for repeated experiments. Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD, eds. *Proc. 2015 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 403–413.

Hart JD (1991) Kernel regression estimation with time series errors. *J. Roy. Statist. Soc. Ser. B* 53(1):173–187.

Hastie T, Tibshirani R, Friedman J (2001) *The Elements of Statistical Learning* (Springer, New York).

Jacod J, Protter PE (2003) *Probability Essentials* (Springer Science & Business Media, New York).

Li KC (1984) Consistency for cross-validated nearest neighbor estimates in nonparametric regression. *Ann. Statist.* 12(1):230–240.

Lin Y, Nelson BL (2016) Simulation analytics for virtual statistics via *k* nearest neighbors. Roeder TMK, Frazier PI, Szechtman R, Zhou E, eds. *Proc. 2016 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 448–459.

Nelson B (2016) 'Some tactical problems in digital simulation' for the next 10 years. *J. Simulation* 10(1):2–11.

Nelson BL, Taaffe MR (2004) The $\mathrm{Ph}_t/\mathrm{Ph}_t/\infty$ queueing system: Part I – The single node. *INFORMS J. Comput.* 16(3):266–274.

Ong KL, Taaffe MR (1989) Nonstationary queues with interrupted Poisson arrivals and unreliable/repairable servers. *Queueing Systems* 4(1):27–46.

Rosenbaum I, Staum J (2015) Database Monte Carlo for simulation on demand. Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD, eds. *Proc. 2015 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 679–688.

Sigman K (1995) *Stationary Marked Point Processes: An Intuitive Approach* (Chapman & Hall, New York).

Smith JS, Nelson BL (2015) Estimating and interpreting the waiting time for customers arriving to a non-stationary queueing system. Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD, eds. *Proc. 2015 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 2610–2621.

Walk H (2010) Strong laws of large numbers and nonparametric estimation. Devroye L, Karasözen B, Kohler M, Korn R, eds. *Recent Developments in Applied Probability and Statistics* (Springer, New York), 183–214.

Wolff RW (1982) Poisson arrivals see time averages. *Oper. Res.* 30(2): 223–231.

Wolff RW (1989) *Stochastic Modeling and the Theory of Queues* (Prentice Hall, Englewoods Cliff, NJ).

Zhao G, Vakili P (2008) Monotonicity and stratification. Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T, Fowler JW, eds. *Proc. 2008 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 313–319.

Zhao G, Borogovac T, Vakili P (2007) Efficient estimation of option price and price sensitivities via structured database Monte Carlo (SDMC). Henderson SG, Biller B, Hsieh M-H, Shortle J, Tew JD, Barton RR, eds. *Proc. 2007 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 984–991.

Zhao G, Zhou Y, Vakili P (2006) A new efficient simulation strategy for pricing path-dependent options. Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM, eds. *Proc. 2006 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 703–710.