Simulating Customer-Averaged Service-Level Metrics in Nonstationary Queueing Systems

- A Queue-Conditional Variance Reduction Method

Yunan Liu

Principal Research Scientist, Supply Chain Optimization Technology, Amazon

Joint work with Ling Zhang

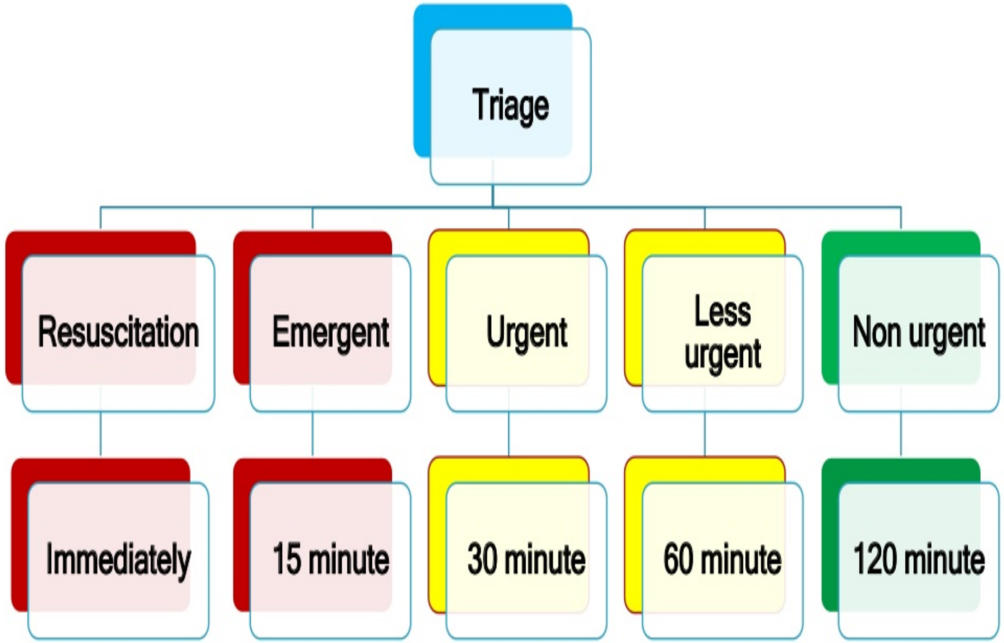
Senior Research Scientist, Last Mile Science, Amazon

WSC 2025

Service-Level Metrics in Practice

- Customer waiting times are critical performance metrics for service quality
- Practical delay-based SL metrics:
  - Mean Waiting Time: $\mathbb{E}[W(t)]$
  - Probability of Delay (PoD): $\mathbb{P}(W(t) > 0)$
    Likelihood customer cannot immediately enter service
  - Probability of Abandonment (PoA): $\mathbb{P}(W(t) > A)$
    Probability an impatient customer leaves before service begins
  - Tail Probability of Delay (TPoD): $\mathbb{P}(W(t) > w)$
    Probability waiting time exceeds target $w > 0$
- Real-World Applications:
  - Emergency Departments:
    Canadian Triage and Acuity Scale sets TPoD targets (15-120 minutes)
  - Call Centers:
    Answer 80% of calls within 20 seconds (TPoD target of 0.2)

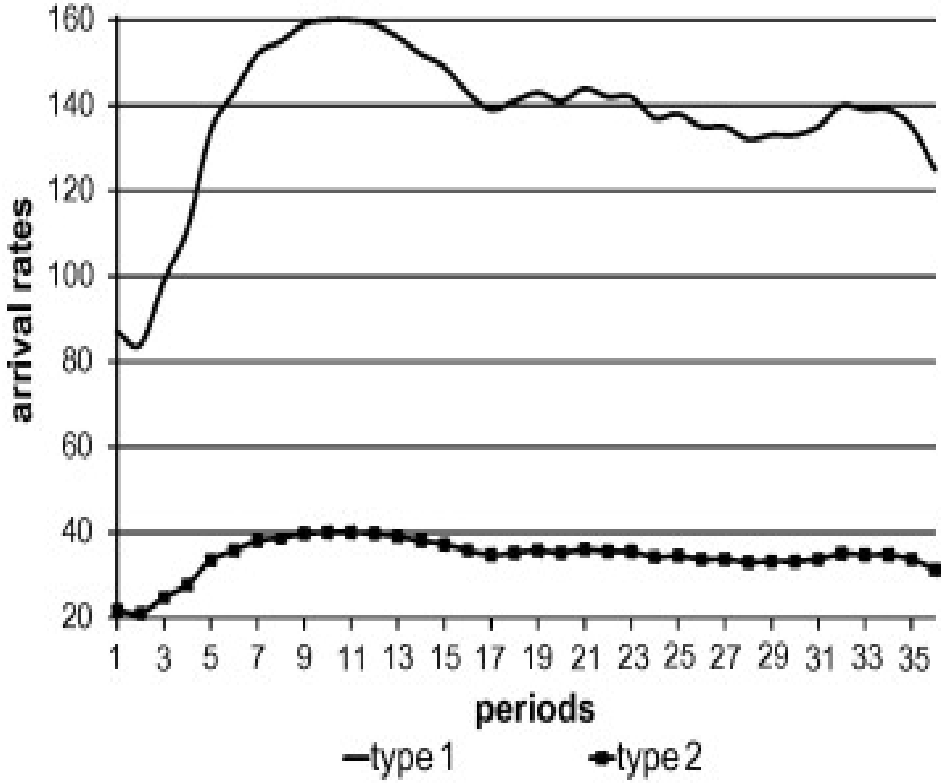
Delay-based Service Levels



Canadian triage and acuity scale (CTAS, Ding et al. 2018)
"CTAS level i patients need to be seen by a physician within w_i minutes $100\alpha_i\%$ of the time", with

$$(w_1, w_2, w_3, w_4, w_5) = (0, 15, 30, 60, 120),$$
$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (0.98, 0.95, 0.9, 0.85, 0.8).$$

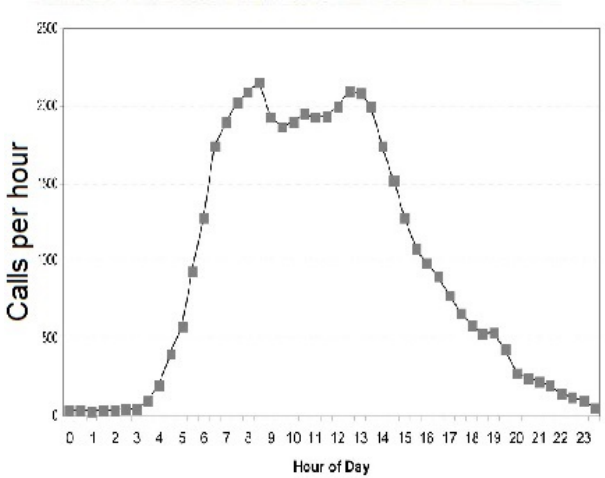
Delay-based Service Levels

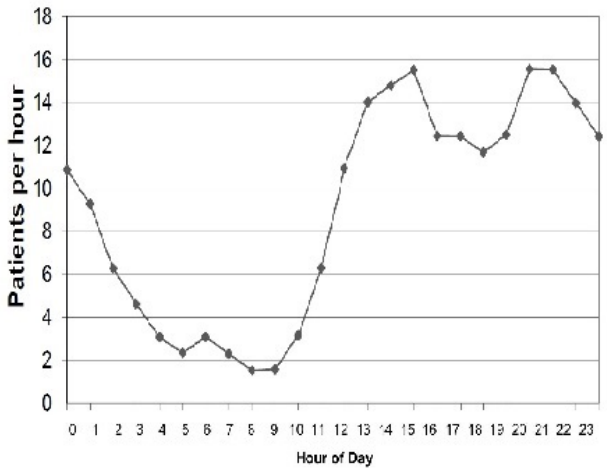


- 80% of type 1 calls need to be answered within 20 seconds ("80-20 rule")
- 50% of type 2 calls need to be answered within 60 seconds
- How many servers are needed over the course of day?
- How to assign a newly idle agents to one of these queues?

Challenge: Nonstationary Queueing Systems

- Real-world systems exhibit significant **time-varying demand**
- Traditional stationary analysis methods become inadequate
- Customer experiences vary significantly throughout the day





Customer-Averaged Service Experience (CASE)

CASE: General Formulation

$$\beta_u \equiv \mathbb{E}\left[\frac{1}{N(T)}\sum_{i=1}^{N(T)} u(W_i)\right]$$

- $N(T)$: Total number of customer arrivals in $[0, T]$
- W_i: Waiting time[1] of i^{th} customer
- $u(\cdot)$: Utility function mapping waiting time to SL metric

Special Cases

- **Average waiting time:** $u(x) = x$
- **Probability of delay:** $u(x) = \mathbf{1}_{\{x>0\}}$
- **Tail Probability of delay:** $u(x) = \mathbf{1}_{\{x>w\}}$

[1] Potential waiting time or actual waiting time

Crude Monte Carlo (CMC)

Algorithm 0: CMC Estimator

1. Simulate nonstationary queueing system over $[0, T]$
2. Record waiting times W_i for all customers
3. Compute $\frac{1}{N(T)} \sum_{i=1}^{N(T)} u(W_i)$
4. Repeat for multiple MC trials and average

Challenges with CMC:

- A bottom-up discrete-event simulation
- High implementation complexity for nonstationary systems
- Requires generating high-granularity customer-level events
- Large number of MC trials needed for tight confidence intervals
- Computationally expensive due to high variance

Our Contributions

Two Proposed Methods:

1. **Q-CASE:** Queue-based conditional estimator
2. **G-CASE:** Gaussian approximation of Q-CASE

Key Innovations

- Conditioning on queue length to reduce variance
- Gaussian approximation for additional efficiency
- Significant variance reduction compared to CMC
- Easier implementation than direct simulation
- Significant end-to-end runtime improvement

The $M_t/M/n_t + M$ Queueing Model

System Components:

- M_t: Poisson arrival with nonstationary rate $\lambda(t)$
- M: Exponential service times with rate μ, $0 < \mu < \infty$
- n_t: Time-varying number of servers $n(t)$
- $+M$: Exponential abandonment time with rate θ, $0 \leq \theta \leq \infty$[2]

Goal: Simulate CASES over entire horizon $t \in [0, T]$.

Potential model generalizations:

- $+GI$ abandonment (immediate)
- GI service (possible, need some work)
- G arrival (need more work)

[2]We cover the special case: (i) no abandonment with $\theta = 0$ and (ii) loss model with $\theta = \infty$.

Queue-based Conditional Estimator (Q-CASE)

- Key Idea:

  Condition on the queue-length process $\mathbf{Q} \equiv \{Q(t), 0 \leq t \leq T\}$ to remove extraneous randomness beyond queue dynamics.

- Pointwise Queue-conditional SL:

  For customer arriving at time t observing queue length q:

$$\widehat{\beta}_u(t, q) \equiv \mathbb{E}[u(W(t))|Q(t) = q]$$

- CASE as weighted average in time:

$$\beta_u = \mathbb{E}\left[\widehat{\beta}_u(\mathcal{T}, Q(\mathcal{T}))\right] = \mathbb{E}\left[\int_0^T f(t)\widehat{\beta}_u(t, Q(t))dt\right]$$

  where $\mathcal{T} \sim f(t) = \frac{\lambda(t)}{\int_0^T \lambda(s)ds}$ is an arbitrary customer's arrival time.

  - From individual to population
  - From customer average to time average

Computing Pointwise SL $\widehat{\beta}_u(t,q)$

Phase-Type Distribution Approach:[3]

- Model waiting time as phase-type distribution
- Pure-death process for queue position evolution
- State space: $\mathcal{S} \equiv \{n-1, n, \ldots, n+q-1, n+q\}$
- Initial state: $n+q$, Absorbing state: $n-1$
- Waiting time: $W(n,q) \stackrel{d}{=} D_q + D_{q-1} + \cdots + D_1 + D_0$, $D_i \sim \text{Exp}(i\theta + n\mu)$
- Waiting time distribution:

$$F(w; n, q) = \mathbb{P}(W(n,q) \leq w) = 1 - \alpha e^{\mathbf{S}w}\mathbf{e}$$

Laplace Transform Approach:

$$\mathcal{L}_{W(n,q)}(s) = \prod_{k=0}^{q} \mathcal{L}_{D_k}(s) = \prod_{k=0}^{q} \frac{n\mu + k\theta}{n\mu + k\theta + s}$$

[3] Konrad and Liu. Real-Time Estimation for the Waiting Time Distributions in Time-Varying Queues. WSC 2023

Algorithm 1: Q-CASE

Algorithm 1: Q-CASE Estimator

1. Precompute $\widehat{\beta}_u(t, q)$ table for all relevant q
2. For each MC trial $j = 1$ to n:
   1. Simulate queue-length process $\{Q_j(t), 0 \leq t \leq T\}$
   2. Compute weighted sum: $\theta^j = \sum_{i=1}^{m} p(t_i)\widehat{\beta}_u(t_i, Q_j(t_i))$
3. Output sample mean: $\widehat{\Theta}(n) = \frac{1}{n}\sum_{j=1}^{n} \theta^j$

- Variance reduction via conditioning on queue length
- Simpler implementation:
  - focuses on system-level queue length only without customer-level events
  - for example: uniformization method
- Precomputation of $\widehat{\beta}(t, q)$ reusable across trials ($\widehat{\beta}$ is independent with $\lambda(t)$)
- Achieving the same accuracy as CMC

Can we do better?

Approximating Queue Length: Fluid Limit

Many-Server Heavy-Traffic (MSHT):
A sequence of $M_t/M/s_t + M$ models indexed by η:

- Arrival rate: $\lambda_\eta(t) = \eta\lambda(t)$
- Service capacity: $n_\eta(t) = \eta n(t)$
- Unscaled: μ, θ

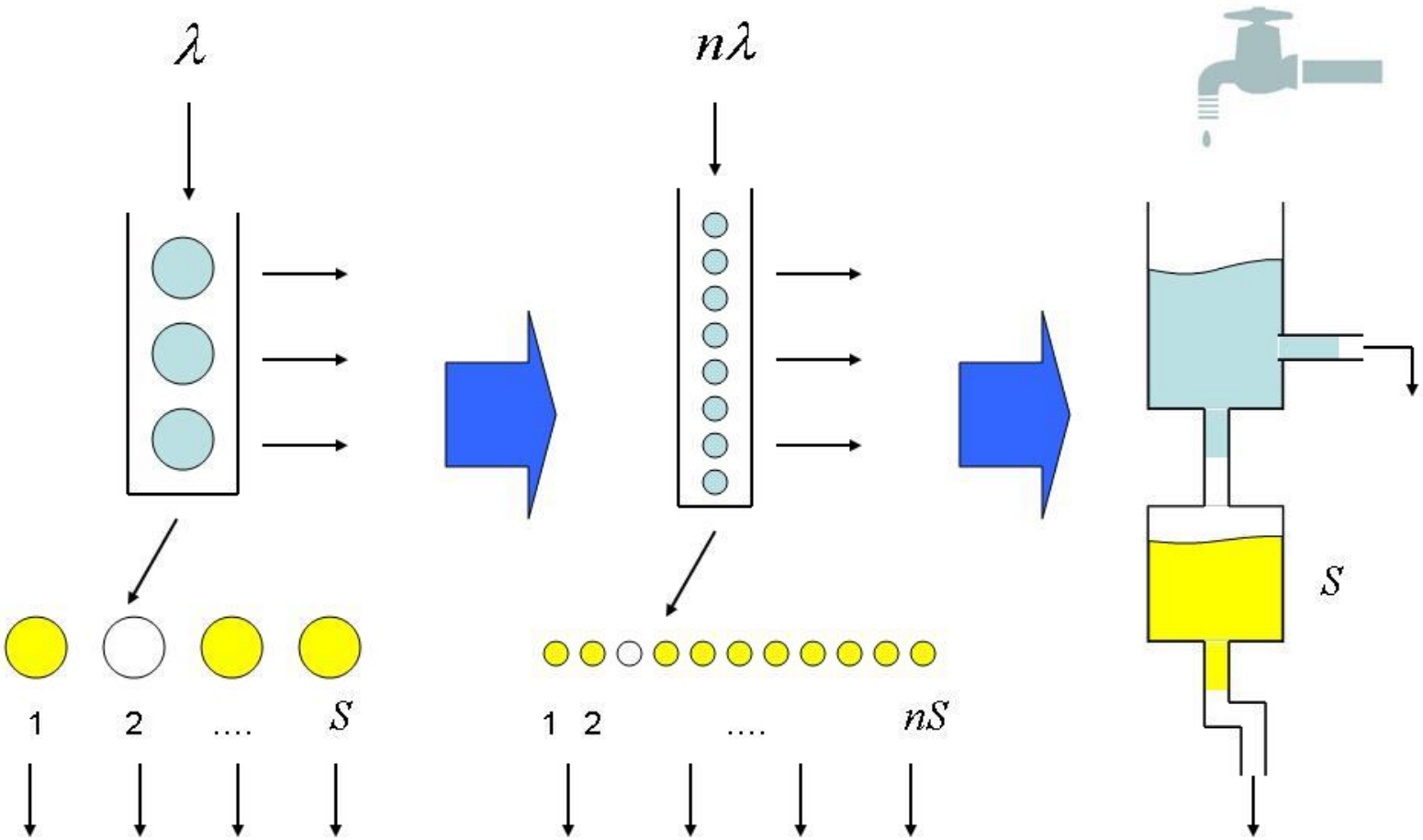
Fluid Limit (FWLLN)

$$X_\eta(t)/\eta \Rightarrow x(t) \quad \text{as } \eta \to \infty$$

where $x(t)$ solves *ordinary differential equation* (ODE):

$$x'(t) = \lambda(t) - \mu \cdot b(t) - \theta \cdot q(t)$$
$$b(t) = x(t) \wedge n(t), \quad q(t) = (x(t) - n(t))^+.$$

Approximating Queue Length: Fluid Limit



Approximating Queue Length: Diffusion Limit

Diffusion Limit (FCLT)

$$(X_\eta(t) - \eta x(t))/\sqrt{\eta} \Rightarrow \widehat{X}(t) \quad \text{as } \eta \to \infty$$

where $\widehat{X}(t)$ is zero-mean Gaussian process.

Queue length approximation:

$$X_\eta(t) \overset{d}{\approx} \eta x(t) + \sqrt{\eta}\widehat{X}(t) \overset{d}{=} \mathcal{N}\left(\eta x(t), \eta\sigma^2(t)\right)$$

$$Q_\eta(t) \overset{d}{\approx} \mathcal{N}\left(\eta x(t) - n_\eta(t), \eta\sigma^2(t)\right)^+$$

$$B_\eta(t) \overset{d}{\approx} \mathcal{N}\left(\eta x(t), \eta\sigma^2(t)\right) \wedge n_\eta(t).$$

Gaussian Approximations

Diffusion limit $\widehat{X}(t)$ solves stochastic differential equation (SDE):[4]

$$\begin{aligned} d\widehat{X}(t) = &- \left[\mu \mathbf{1}(x(t) < n(t)) + \theta \mathbf{1}(x(t) > n(t))\right] \widehat{X}(t) \\ &- d\mathcal{B}_\lambda \left(\int_0^t \lambda(s) ds \right) - d\mathcal{B}_a \left(\int_0^t \theta (x(s) - n(s))^+ ds \right) \\ &- d\mathcal{B}_s \left(\int_0^t \mu (x(s) \wedge n(s)) ds \right) \end{aligned}$$

where $\mathcal{B}_\lambda$, $\mathcal{B}_a$ and $\mathcal{B}_s$ are independent BM's accounting for the asymptotic variability of the arrival, abandonment and service processes.

Variance process:

$$\begin{aligned} \frac{d\sigma^2(t)}{dt} = &-2 \left[\theta \mathbf{1}(x(t) > n(t)) + \mu \mathbf{1}(x(t) < n(t))\right] \sigma^2(t) \\ &+ \lambda(t) + \theta (x(t) - n(t))^+ + \mu (x(t) \wedge n(t)). \end{aligned}$$

[4] Mandelbaum et al. (1998) Strong approximations for Markovian service networks.

G-CASE: Gaussian Approximation Estimator

G-CASE Formulation:

$$\begin{aligned}\beta^G &\equiv \mathbb{E}\left[\widehat{\beta}\left(\mathcal{T}, (x(\mathcal{T}) - n(\mathcal{T}) + \sigma(\mathcal{T})\mathcal{Z})^+\right)\right] \\ &= \mathbb{E}\left[\int_0^T f(t)\widehat{\beta}\left(t, (x(t) - n(t) + \sigma(t)\mathcal{Z})^+\right) dt\right],\end{aligned}$$

where $\mathcal{T}$ is an arbitrary customer's arrival time, $\mathcal{Z} \sim \mathcal{N}(0,1)$.

Antithetic Variables:

$$\beta_A^G \equiv \mathbb{E}\left[\int_0^T f(t)\frac{\widehat{\beta}\left(t, (x(t) - n(t) + \sigma(t)\mathcal{Z})^+\right) + \widehat{\beta}\left(t, (x(t) - n(t) + \sigma(t)(-\mathcal{Z}))^+\right)}{2} dt\right]$$

Note: $\widehat{\beta}(t, q(t)) = \mathbb{E}[W(t)|Q(t) = q(t)]$ is increasing in $q(t)$.

Algorithm 2: G-CASE with Antithetic Variables

Algorithm 2: G-CASE Estimator

1. **For each MC trial** $j = 1$ to n:
   1. Generate $\mathcal{Z}_j \sim \mathcal{N}(0,1)$
   2. Solve ODE's for $x(t)$ and $\sigma^2(t)$
   3. For each time point t_i:
      - Set $q_1 = (x(t_i) - n(t_i) + \sigma(t_i)\mathcal{Z}_j)^+$, $q_2 = (x(t_i) - n(t_i) - \sigma(t_i)\mathcal{Z}_j)^+$
      - Compute $\widehat{\beta}_j(t_i, q_1)$ and $\widehat{\beta}_j(t_i, q_2)$
   4. Compute trial estimate: $\theta^j = \sum_{i=1}^m p(t_i) \frac{\widehat{\beta}_j(t_i,q_1)+\widehat{\beta}_j(t_i,q_2)}{2}$
2. **Output:** Sample mean $\widehat{\Theta}(n) = \frac{1}{n}\sum_{j=1}^n \theta^j$

Key Advantages:

- Only need to generate single Gaussian variable per trial
- No need for full queue-trajectory simulation
- No discrete events needed
- Significant variance reduction

Numerical Experiments

Experiment setting:

- $M_t/M/s_t + M$ queue with sinusoidal arrival rate
- $\lambda(t) = n(1 + 0.2\cos(t))$
- Staffing function: n servers
- Service rate: $\mu = 1$
- Abandonment rate: $\theta = 0.5$
- System scale: $n = 1000$ servers
- MC trials: 1000 for all estimators

Two CASE Performance Metrics:

1. Customer-averaged waiting time: $u(x) = x$
2. Fraction with waiting times $\leq w$: $u(x) = \mathbf{1}_{\{x \leq w\}}$, $w = 0.5$

Confidence Intervals

	Time	$t = 8$	$t = 10$	$t = 16$	$t = 20$
Customer-averaged waiting time	CMC	0.044 ± 3e-03	0.045 ± 3e-03	0.056 ± 2e-03	0.058 ± 2e-03
	Q-CASE	0.045 ± 1e-03	0.046 ± 1e-03	0.057 ± 1e-03	0.059 ± 1e-03
	G-CASE	0.043 ± 8e-06	0.044 ± 4e-05	0.055 ± 4e-05	0.057 ± 4e-05
Fraction of waiting times below a target	CMC	0.223 ± 2e-02	0.221 ± 2e-02	0.275 ± 1e-02	0.284 ± 1e-02
	Q-CASE	0.228 ± 5e-03	0.23 ± 6e-03	0.286 ± 5e-03	0.293 ± 5e-03
	G-CASE	0.221 ± 2e-03	0.224 ± 1e-03	0.282 ± 2e-03	0.288 ± 2e-03

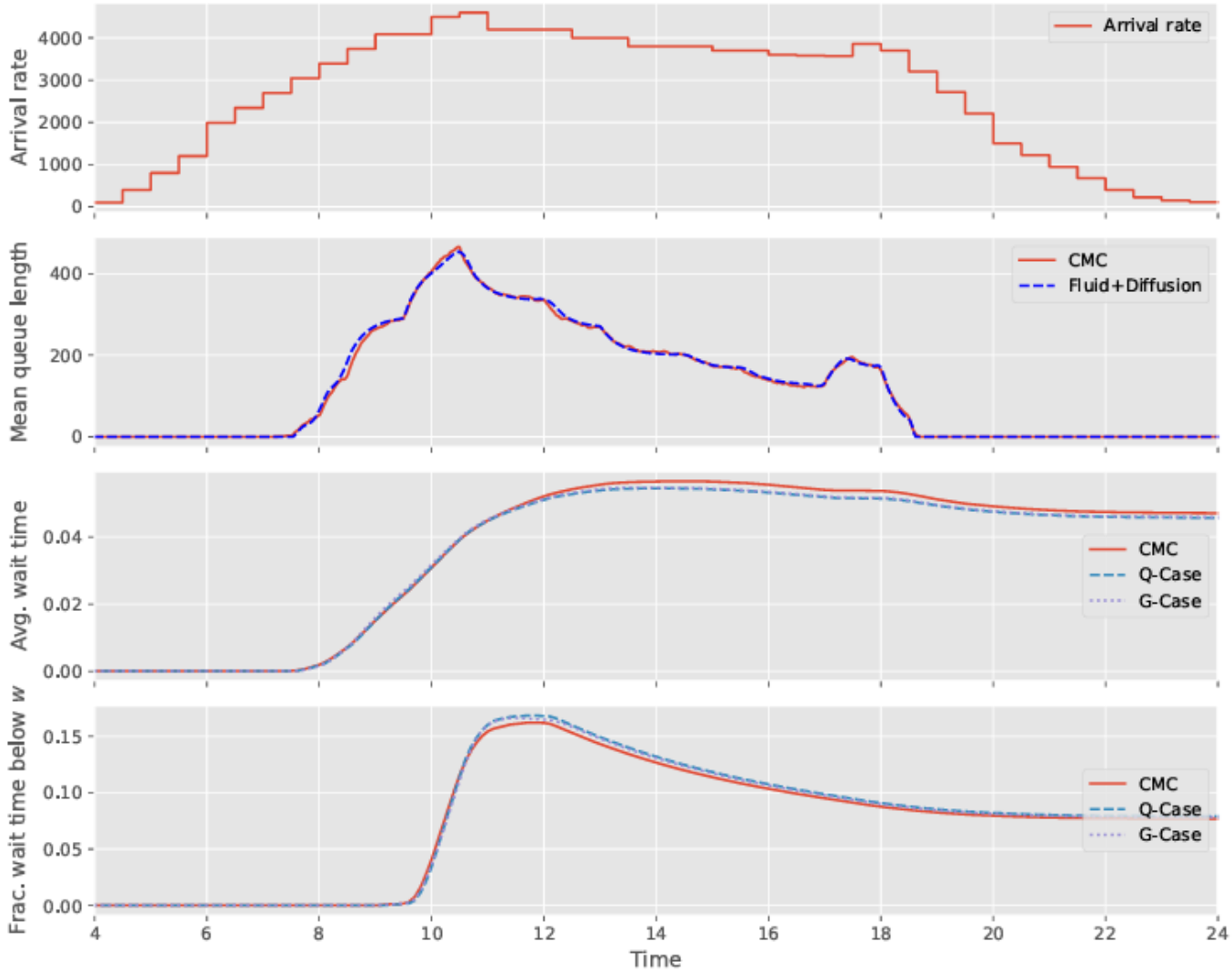
- Q-CASE vs. CMC:

  A smaller CI by eliminating additional variability beyond the queueing length.

- G-CASE vs. Q-CASE:

  A further smaller CI by reducing path-level variability of queue length (additional VR due to antithetic variable).

Realistic Call Center Arrival Pattern



- $T = 24$ hours, $n = 400$ servers
- Mean service time: 10 minutes, mean abandonment time: 20 minutes

Computational Efficiency

Runtime Comparison:

- **CMC:** $\sim$1 minute 40 seconds
- **G-CASE:** $\sim$0.3 seconds
- **Speedup:** $\sim$333$\times$ faster

Efficiency Factors:

- G-CASE requires only single Gaussian variable per trial
- CMC requires complete bottom-up discrete-event simulation
- Precomputation of $\widehat{\beta}(t, q)$ values for efficiency
- Antithetic variables provide additional variance reduction

Insights

Conditioning Principle:

- Separates queue-length randomness from waiting-time randomness given queue
- Queue-length process captures essential system dynamics
- Waiting-time distribution can be analyzed conditional on queue state
- Reduces dimension of stochastic variability

Implementation Considerations:

- Precomputation of $\widehat{\beta}(t, q)$ tables for efficiency
- Numerical methods for phase-type distributions
- ODE solvers for fluid and variance equations
- Parallel implementation for multiple MC trials

Limitations and Extensions

Current Limitations:

- Focus on Markovian systems ($M_t/M/s_t + M$)
- Assumes exponential service and abandonment times
- Gaussian approximation accuracy for small systems
- Requires solving ODEs for fluid and variance

Future Extensions:

- Non-Markovian systems $G_t/GI/s_t + GI$
  - Qplex?[5] transfer learning?[6]
- Multi-class queueing systems, network queueing systems
- Correlation in time: Given SL in $[0, 2]$, simulate SL in $[4, 6]$.

[5] Dieker and Hackman. QPLEX: A Computational Modeling and Analysis Methodology for Stochastic Systems.

[6] Garyfallos and Liu. Solving Nonstationary Non-Markovian Queueing Systems: A Transfer Learning Neural Network Approach.

Thank You!

Questions?

yunanliu@amazon.com

https://yliu48.github.io/

Waiting Time Definitions

Potential Waiting Time (PWT)

- Offered waiting time assuming customer remains indefinitely patient
- Independent of customer abandonment behavior
- Key metric for assessing system workload

Actual Waiting Time (AWT)

- Total time customers spend in queue (regardless of abandonment)
- More directly observable from real-world data
- More relevant for customer experience

Our approach handles both PWT and AWT!