Predictable Variability = "Piecewise Stationary" load
Goal: Predictably Stable Performance, but HOW?

Arrivals

Queues

Waiting
Staffing Time-Varying Queues:

Two Common Approaches:

SSA – Simple Stationary Approximation.
Constant staffing levels, based on steady-state M/M/N, with $\lambda =$ long-run average number of arrivals.

PSA – Point-wise Stationary Approximation.
Time-varying staffing levels, based on steady-state M/M/N, with $\lambda = \lambda(t)$ at each time $t$.

Could result in time-varying (highly oscillating) performance (utilization, service), which is undesirable.
Simple Stationary Approximation (SSA, \( \alpha=0.2 \))
Point-wise Stationary Approximation (PSA, $\alpha=0.2$)
Example: "Real" Call Center

Two-hump arrival functions are common
(Adapted from Green L., Kolesar P., Soares J. for benchmarking.)

Assume: Service and abandonment rates are both
exponential having mean 0.1 (6 min.)
QD Staffing ($\alpha=0.1$)

ED Staffing ($\alpha=0.9$)
QED Staffing ($\alpha=0.5$)
Congestion (Queue, Wait)

QD
$\alpha = 0.1$
Negligible

QED
$\alpha = 0.5$
Seconds

ED
$\alpha = 0.9$
Minutes

ED
$\bar{I} = 0.9$ Minutes

QD
$\bar{I} = 0.1$ Seconds

QED
$\bar{I} = 0.5$ Seconds

ED
$\bar{I} = 0.5$ Minutes
Erlang-A: Moderate (Im)patience

- M/M/N + M queue, with
  
  service rate μ equals θ abandonment rate

- Lt: number-in-system at time t (Birth & Death)

- For any N, transition-rates for \{Lt, t ≥ 0\}:

  Note: The same transition rates as \textbf{M/M/∞}

\[
\begin{align*}
\lambda &\quad \lambda \\
0 &\quad 1 &\quad 2 &\quad \cdots \cdots &\quad N-1 &\quad N &\quad N+1 &\quad \cdots \cdots \\
\mu &\quad 2\mu &\quad \cdots \cdots &\quad N\mu &\quad N\mu + \theta &\quad (N+1)\mu
\end{align*}
\]
Square-Root Staffing: Motivation

\[ P\{W_q (M / M / N + M) > 0\} = \text{PASTA} \]
\[ P\{L(M / M / N + M) \geq N\} = \theta = \mu \]
\[ P\{L(M / M / \infty) \geq N\} \]

Fact: \( L(M / M / \infty) \sim \text{Poisson}(R) \); \( R = \frac{\lambda}{\mu} \) offered load

For \( R \) not too small:
\[
\frac{d}{d} \text{L}(M/M/\infty) \approx \text{Normal}(R,R) = R + Z\sqrt{R}
\]

\[ \Rightarrow \quad P\{W_q > 0\} \approx P\left\{ Z \geq \frac{N-R}{\sqrt{R}} \right\} = 1 - \phi\left(\frac{N-R}{\sqrt{R}}\right) \]

Given target delay-probability \( \alpha = 1 - \phi\left(\frac{N-R}{\sqrt{R}}\right) \)

\[ \Rightarrow \quad N = R + \beta \cdot \sqrt{R}, \quad \text{with} \quad \beta = \phi^{-1}(1-\alpha) \]

\( N \) is the "least integer for which" \( P\{W_q > 0\} \leq \alpha \)
Time-Varying Arrivals

Extension: \( M_t / M / N_t + M \) \( (\mu=\theta) \)

\[ N_t = R_t + \beta \cdot \sqrt{R_t} \]

Fact: \( L_t \sim \text{Poisson}(R_t) \)

\( R_t \) – the offered load at time \( t \), namely:

\[ R_t = E\lambda(t - S_e) \cdot E(S) = E \int_{t-S_e}^{t} \lambda(u)du \]

\( S_e \) – excess service

\[ E(S_e) = E(S) \left( \frac{1 + c^2_s}{2} \right) \]
Time-Varying Arrivals

Extension: \( M_t / M / N_t + M \) (\( \mu=\theta \))

\[ N_t = R_t + \beta \cdot \sqrt{R_t} \]

Fact: \( L_t \sim \text{Poisson}(R_t) \)

\( R_t \) – the offered load at time \( t \), namely:

\[ R_t = E\lambda(t - S_c) \cdot E(S) = E \int_{t-S}^{t} \lambda(u)du \]

\( S_c \) – excess service

\[ E(S_c) = E(S)\frac{1 + c_s^2}{2} \]

\( L_t \overset{d}{=} N(R_t, R_t) \) \text{ hence, as before:}

\[ N_t = \left[ R_t + \beta \cdot \sqrt{R_t} \right], \quad \beta = \phi^{-1}(1 - \alpha) \]

hopefully yields time-stable delay probability \( \alpha \):

Indeed, but in fact \textbf{TIME-STABLE PERFORMANCE}!

What if \( \mu \neq \theta \)?

Use an \textit{Iterative Algorithm} that is \textit{Simulation-Based}
Performance Measures

- **Delay probability in interval** \( t \), calculated by the fraction of customers who are not served immediately upon arrival, out of all arriving customers during the \( t \) time-interval

- **Average waiting time in interval** \( t \), calculated by the average waiting time of all customers arriving during the \( t \) time-interval.

- **Average queue length in interval** \( t \), taken constant over the time-interval. The queue length is averaged over all replications

- **Tail probability in interval** \( t \), calculated as the probability that queue size equals or exceeds some threshold (e.g. 3 times average queue)

- **Servers' Utilization in interval** \( t \), calculated as the fraction of busy-servers during a time-interval (accounting for servers who are busy only a fraction of the interval)

- **Service grade** \( \beta_t \) in interval \( t \), which arises from the following "Square-Root Staffing" rule:

\[
N_t = R_t + \beta_t \sqrt{R_t}
\]
Two-hump arrival functions are typical
(Adapted from Green L., Kolesar P., Soares J. for benchmarking.)

- Service and abandonment rates are both exponential having mean 0.1 (6 min.)
QED Staffing ($\alpha=0.5$)
Erlang-A: Theoretical vs. Empirical

\[ P\{\text{Wait}>0\} = \alpha \text{ vs. } \beta \quad (N=R+\beta\sqrt{R}) \]

Moderate Patience

![Graph showing the comparison between theoretical and empirical data for the probability of wait time exceeding zero. The graph plots \( P\{\text{Wait}>0\} \) against \( \beta \) for different values of \( \alpha \). The results are compared with empirical data and GMR predictions.]
GMR(x) describes the asymptotic probability of delay as a function of $\beta$ when $\frac{\theta}{\mu} = x$. Here, $\theta$ and $\mu$ are the abandonment and service rate, respectively.
Erlang-A: $P\{\text{Abandon}\}^*\sqrt{N}$ vs. $\beta$
Iterative Algorithm

Inputs

- System primitives:
  - arrival function, service-time distribution, patience distribution (when relevant);
- Target delay probability $\alpha$;
- Time horizon $[0,T]$.

Outputs

- Staffing function, aiming at a delay probability $\alpha$ is over $[0,T]$.

Starting point: The infinite-server heuristics by Jennings, M., Massey, Whitt (1996)
Algorithm (cont.)

**Notation:** \( \forall t \in [0,T] \) (practically \( t=0, \Delta, 2\cdot\Delta, \ldots \))

- \( N_i(t) \) – staffing level at time \( t \), determined in iteration \( i=1,2,\ldots \)
- \( L_i(t) \) – number in the system at \( t \), under staffing function \( s_i(t) \).

**Algorithm:**

1. \( i=0; N_0(t) \equiv 0 \) (delay probability =0)
2. Evaluate the distribution of \( L_i(t) \), using simulation.
3. Determine \( N_{i+1}(t) \) as follows:
   \[
   N_{i+1}(t) = \arg \min \{ c : P\{L_i(t) \geq c\} < \alpha \}, \quad 0 \leq t \leq T.
   \]
4. Check stopping condition:
   - if \( \|N_{i+1}(\cdot) - N_i(\cdot)\|_{\infty} \leq 1 \), then \( N_{i+1}(\cdot) \) is our staffing level;
   - else \( i := i+1 \), and go back to (2).

\( \infty \) Last iteration. The algorithm converges to a Staffing Function \( N_{\infty}(\cdot) \) least for which

\[
P\{L_{\infty}(t) \geq N_{\infty}(t)\} \leq \alpha, \quad 0 \leq t \leq T.
\]
WHY SERVICE STINKS

Companies know just how good a customer you are—and unless you’re a high roller, they would rather lose you than fix your problem.

PAGE 118