Dynamically Scheduling and Maintaining a Flexible Server

Jefferson Huang

Operations Research Department
Naval Postgraduate School

INFORMS Annual Meeting
November 7, 2018

Co-Authors:
Douglas Down (McMaster), Mark Lewis (Cornell), Cheng-Hung Wu (NTU)
Scheduling and Maintenance

A **flexible** server/machine can handle different types of **jobs**.
  ▶ e.g., different kinds of customers, different products

The service capacity/rate of the server can **deteriorate** over time.
  ▶ e.g., fatigue, wear & tear, needs cleaning

**Questions:**

1. How should the server’s effort be allocated (i.e., **scheduled**)?

2. When should the server be **maintained**?

We consider these questions in the context of a **queueing** system.
Queue with State-Dependent Service Rates

Consider an \textit{M/M/1} queueing system with two arrival classes.

For class $i = 1, 2$,

- arrival rate is $\lambda_i$
- holding cost rate is $c_i$

The service rates depend on the server state $s \in \{1, \ldots, S\}$.

- $\mu^s_i = \text{class } i \text{ service rate when server state is } s$

The server state evolves according to a continuous-time Markov chain.

- jump probabilities $J_{s,t}, s, t \in \{1, \ldots, S\}$
- holding time rates $\alpha_s, s \in \{1, \ldots, S\}$
Cai, Hasenbein, Kutanoglu & Liao (2013) consider a closely related 2-class model, with a different cost, service, and degradation structure.

Other work in joint service/production and maintenance:

- **Non-Queueing**: Yao, Xie, Fu & Marcus (2005), Iravani & Duenyas (2002), Sloan & Shanthikumar (2000)
Scheduling

For now, assume we only need to decide how to allocate the server.

- At each decision epoch (arrival, service completion, server state change), decide which class to serve.

A policy for doing this can depend on the current queue lengths and server state, as well as the history (past queue lengths, server states, and decisions).

- $Q_i^\pi(t) =$ number of class $i$ jobs at time $t$, under policy $\pi$

**Objective:** Find a policy $\pi$ minimizing the long-run expected average cost

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E} \int_0^T [c_1 Q_1^\pi(t) + c_2 Q_2^\pi(t)] \, dt$$
Scheduling

**Definition**

The *c_μ*-Rule is the scheduling policy where

\[
\text{server state is } s \implies \text{prioritize class } i^* \in \arg \max_{i=1,2} c_i \mu_i^s
\]

**Well-Known:** If the server state does not change, then the *c_μ*-Rule is optimal. (Buyukkoc, Varaiya & Walrand 1985)

**Question:** Is the *c_μ*-Rule optimal when the server state changes?
Suboptimality of the $c\mu$-Rule

**Example:**
- arrival rates $\lambda_1 = 5$, $\lambda_2 = 0.8$
- cost rates $c_1 = c_2 = 1$
- $S = 2$ server states

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1^1 = 10$</td>
<td>$\mu_2^1 = 1$</td>
</tr>
<tr>
<td>$\mu_1^2 = 10$</td>
<td>$\mu_2^2 = 2$</td>
</tr>
</tbody>
</table>

$\alpha_1 = 1$, $J_{12} = 1$
$\alpha_2 = 1$, $J_{21} = 1$
$c_1 \mu_1^1 = 10 > 1 = c_2 \mu_1^1$
$c_1 \mu_1^2 = 10 > 2 = c_2 \mu_2^1$

The $c\mu$-Rule (always prioritize class 1) leads to an **infinite average cost**!
- (long-run fraction of time busy with class 1) $= \frac{\lambda_1}{10} = \frac{1}{2}$
- (average class 2 service rate) $= \frac{1}{2} (\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2) = 0.75 < 0.8 = \lambda_2$

The $c\mu$-rule is **not optimal**, because the following policy leads to a **finite average cost**:
- If the server state is $s$, prioritize class $s$. 
When is the $c\mu$-Rule optimal?

**Theorem**

Suppose

\[ \mu_{1}^{s-1}\mu_{2}^{s} = \mu_{1}^{s}\mu_{2}^{s-1} \quad \forall s > 1. \]  

(1)

Then the $c\mu$-Rule is optimal.

- (1) means that the ratio between the service rates is constant in $s$:

\[ \frac{\mu_{1}^{s-1}}{\mu_{2}^{s-1}} = \frac{\mu_{1}^{s}}{\mu_{2}^{s}} \]

- Under (1), a variant of the interchange argument in (Nain 1989) can be used to prove the Theorem.
Scheduling and Maintenance

Same \( M/M/1 \) model as before, with the following modifications:

- Additional server state 0 (server is down for maintenance)
- \( 0 = \mu_i^0 < \mu_i^1 \leq \cdots \leq \mu_i^S \) for \( i = 1, 2 \)
  - \( s \) = condition of the server

- **Preventive Maintenance (PM)** when \( s > 0 \)
  - Send the server to state 0
  - Incur cost \( K_{PM} \)
  - Maintenance time has general distribution \( G \)

- **Deterioration** when \( s > 0 \)
  - Server transitions from state \( s \) to \( s - 1 \) at rate \( \alpha_s \)
  - If an uncontrolled transition to server state 0 occurs, the **Corrective Maintenance (CM)** cost \( K_{CM} \) is incurred.
Scheduling and Maintenance

A policy stipulates, given the current queue lengths, server state, and history of the process, whether to

- initiate preventive maintenance, or
- serve one of the classes.

For a policy $\pi$,

- $Q_i^\pi(t) =$ number of class $i$ jobs at time $t$, under $\pi$
- $M_{PM}^\pi(t) = \begin{cases} 1 & \text{if PM is initiated at time } t \text{ under } \pi \\ 0 & \text{otherwise} \end{cases}$
- $M_{CM}^\pi(t) = \begin{cases} 1 & \text{if CM is initiated at time } t \text{ under } \pi \\ 0 & \text{otherwise} \end{cases}$
- $t_n^\pi =$ time of the $n^{th}$ maintenance initiation, under $\pi$

**Objective:** Find a policy $\pi$ minimizing the long-run expected average cost

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{n: t_n^\pi \leq T} [K_{PM} M_{PM}^\pi(t_n^\pi) + K_{CM} M_{CM}^\pi(t_n^\pi)] + \int_0^T \sum_{i=1}^2 c_i Q_i^\pi(t) \, dt \right]$$
Structure of Optimal Policies

**Theorem**

Suppose

\[ \mu_1^{s-1} \mu_2^s = \mu_1^s \mu_2^{s-1} \quad \forall s > 1, \]

and that there exists a server state \( s^* \) such that

\[ \frac{\lambda_1}{\sum_{s=s^*}^{S} (\mu_1^s / \alpha_s)} + \frac{\lambda_2}{\sum_{s=s^*}^{S} (\mu_2^s / \alpha_s)} < \frac{1}{(1/\alpha_0)} + \sum_{s=s^*}^{S} (1/\alpha_s). \]

Then there is an optimal policy that

(i) schedules according to the \( c\mu \)-Rule, and

(ii) makes maintenance decisions monotonically in the server state.

- “schedules according to the \( c\mu \)-Rule” means:
  - If the policy says to serve a class (rather than do preventive maintenance), use the \( c\mu \)-Rule to select which one.

- “makes maintenance decisions monotonically in the server state” means that for each fixed number of class 1 jobs and number of class 2 jobs in the system,
  - maintain when server state is \( s \) \( \implies \) maintain when it is \( s-1 \)
Structure of Optimal Policies

Example: $c\mu$-Rule says to prioritize class 2:
Conclusions

We considered a combined scheduling and maintenance problem for a queueing system.

**Key Takeaways:**

- The $c\mu$-Rule can be very bad.
- If degradation reduces the service rates by the same percentage, then attention can be restricted to policies that
  - schedule according to the $c\mu$-Rule, and
  - call for maintenance monotonically in the server state.

Regarding the structure of optimal or near-optimal policies, *the picture is still very incomplete*.

- Heavy-traffic approximations?
- One-step policy improvement?