The Role of Information in Stability of Queueing Systems

Azam Asanjarani (joint work with Dr. Yoni Nazarathy)
School of Mathematics and Physics, The University of Queensland

Introduction

We address the problem of stabilizing control for complex queueing systems with known parameters but unobservable Markovian random environment. In such systems, the controller needs to assign servers (e.g., communication channels, transmitters, manufacturing machines) to units requiring processing (e.g., file transfers, widgets). We explore the role of information in the system stability region.

<table>
<thead>
<tr>
<th>Observation Y(t)</th>
<th>Policy π</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Observation</td>
<td>( (X(t), X(t)) )</td>
</tr>
<tr>
<td>State Observation</td>
<td>( X_{N(t)}(t) )</td>
</tr>
<tr>
<td>Unusual Observation</td>
<td>( E(Y_{N(t)}(t)) )</td>
</tr>
<tr>
<td>Queue Observation</td>
<td>( E(t) )</td>
</tr>
<tr>
<td>No Observation</td>
<td></td>
</tr>
</tbody>
</table>

Goal

Design a stable system, i.e., an efficient queueing system that reduces the number of units requiring processing and keeps it stochastically bounded under an on-line controller.

System Model

The system is operating in discrete time steps \( t = 0, 1, \ldots \) according to the following chain of actions:

1. New arrival \( \epsilon_{j} \) and \( \mu_{j} \) arrival
2. Server \( \epsilon_{j} \) and \( \mu_{j} \) depart
3. Server \( \epsilon_{j} \) and \( \mu_{j} \) stay
4. Server \( \epsilon_{j} \) and \( \mu_{j} \) move
5. Server \( \epsilon_{j} \) and \( \mu_{j} \) death

The probability transition matrix of each server environment is:

\[
P = \begin{bmatrix}
q & \bar{q} \\
\bar{q} & q
\end{bmatrix},
\]  

where \( q = \frac{\lambda}{\mu + \lambda} \) and \( \bar{q} = \frac{\mu}{\mu + \lambda} \).

Moreover, we can consider the spread of \( \mu \) of each server as \( \mu_{0} = \gamma_{j} - \epsilon_{j} \) and \( \mu_{1} = \gamma_{j} + \epsilon_{j} \), where \( \epsilon_{j} \) shows the spread.

The probability transition matrix of each server environment is:

\[
P = \begin{bmatrix}
q & \bar{q} \\
\bar{q} & q
\end{bmatrix} = \begin{bmatrix}
\frac{\lambda}{\mu + \lambda} & \frac{\mu}{\mu + \lambda} \\
\frac{\mu}{\mu + \lambda} & \frac{\lambda}{\mu + \lambda}
\end{bmatrix}.
\]

Moreover, we can consider the spread of \( \mu \) of each server as \( \mu_{0} = \gamma_{j} - \epsilon_{j} \) and \( \mu_{1} = \gamma_{j} + \epsilon_{j} \), where \( \epsilon_{j} \) shows the spread.

Conclusion

- We tried to handle control of stochastic systems with partial observations.
- Explicit analysis of such systems is extremely challenging. However, our model and numerical results, pave the way for explicit proofs of the structural properties.
- Insights on the role of information on system stability.

Future Work

- We are going to extend this work to more general server environment models, as well as systems with more queues and control decisions.

Key Idea

Using belief states to summarise the history of observations, \( Y(t - 1), Y(t - 2), \ldots \), into sufficient statistics that are updated by the controller.

\[ \omega_{j}(t) = P[X(t) = 1 \mid \text{Prior knowledge to time } t] \]

Numerical Investigations

Our various numerical experiments indicate and result the following:

- The ordering \( \mu_{0} \leq \mu_{1} \leq \mu_{2} \) holds. See Fig. (I).
- Increasing \( \rho_{1} \) always yields an increase in \( \mu^{*} \). See Fig. (II).
- The stability region grows as the number of controller belief states increases. See Fig. (III).
- Though the myopic policy does not appear to be generally optimal, when both servers are identical, the optimal policy is the symmetric myopic policy. See Fig. (III).
- In all cases, the optimal policy is given by a non-decreasing switching curve. That is, there exists a function \( \omega_{2}(\omega_{1}) \) where the optimal policy is \( \pi(\omega_{1}, \omega_{2}) = \begin{cases} 2 \text{ if } \omega_{2} \geq \omega_{2}(\omega_{1}) \text{ or } 1 \text{ if otherwise.} 
\end{cases} \)
- Increasing spread of servers \( \epsilon_{j} \), always yields an increase in \( \mu^{*} \). See Fig. (III).
- The switching curve for the queue observation case depends on \( \lambda \). Further, when \( \lambda \) is at either of the extreme points (\( \lambda = 0 \) or \( \lambda = 1 \)), the queue observation case agrees with the output observation case. See Fig. (IV).

References


Acknowledgements

This work was supported by the ARC Centre of Excellence for Mathematical and Statistical Frontiers. The authors are supported by Australian Research Council (ARC) grant DE130100291 and would like to thank her supervisor Dr. Yoni Nazarathy.