Methodology and Computing in Applied Probability (2020) 22:949–968
https://doi.org/10.1007/s11009-019-09750-4

The Role of Information in System Stability with Partially Observable Servers

Azam Asanjarani[1] · **Yoni Nazarathy**[2]

Received: 8 February 2018 / Revised: 8 January 2019 / Accepted: 23 October 2019 /
Published online: 16 November 2019

Abstract

We present a methodology for analyzing the role of information on system stability. For this we consider a simple discrete-time controlled queueing system, where the controller has a choice of which server to use at each time slot and server performance varies according to a Markov modulated random environment. At the extreme cases of information availability, that is when there is either full information or no information, stability regions and maximally stabilizing policies are trivial. But in the more realistic cases where only the environment state of the selected server is observed, only the service successes are observed or only queue length is observed, finding throughput maximizing control laws is a challenge. To handle these situations, we devise a Partially Observable Markov Decision Process (POMDP) formulation of the problem and illustrate properties of its solution. We further model the system under given decision rules, using Quasi-Birth-and-Death (QBD) structure to find a matrix analytic expression for the stability bound. We use this formulation to illustrate how the stability region grows as the number of controller belief states increases. The example that we consider in this paper is a case of two servers where the environment of each is modulated like a Gilbert-Elliot channel. As simple as this case seems, there appear to be no closed form descriptions of the stability region under the various regimes considered. However, the numerical approximations to the POMDP Bellman equations together with the numerical solutions of the QBDs, both of which are in agreement, hint at a variety of structural results.

Keywords System stability · POMDP · Control · Queueing systems · Optimal · Bellman equations · QBD · Information · Markov models

Mathematics Subject Classification 2010 60J28 · 49L20

✉ Azam Asanjarani
azam.asanjarani@auckland.ac.nz

[1] The University of Auckland, Auckland, New Zealand

[2] The University of Queensland, Brisbane, Australia

1 Introduction

Performance evaluation and control of queueing systems subject to randomly varying environments is an area of research that has received much attention during the past few decades (see for example, Cecchi and Jacko (2016), Johnston and Vikram (2006), and Sadeghi et al. (2008) and references therein). This is because numerous situations arise in practice where a controller needs to decide how to best utilise resources, and these are often subject to changing conditions. Examples of such situations arise in wireless communication, supply chain logistics, health care, manufacturing and transportation. In all these situations, it is very common for service rates to vary in a not fully predictable manner. Using Markovian random environments has often been a natural modelling choice due to the tractability and general applicability of Markov models. See for example Section A.1 in Meyn (2008) for a general discussion on the ubiquity of Markov models.

The bulk of the literature dealing with performance evaluation and control of these types of problems, has considered the situation where the state of the underlying random environment is observable. In such a situation, it is already a non-trivial task to carry out explicit system performance analysis (see Kella and Whitt (1992) as an example). Further, finding optimal or even merely stabilizing control is typically a formidable achievement (see for example (Baccelli and Makowski 1986; Tassiulas and Ephremides 1993) or the more recent (Halabian et al. 2014)). But in practice, the actual environment state is often not a directly observed quantity, or is at best only partially observable. The situation is further complicated when control decisions do not only affect instantaneous rewards, but also the observation made. In classic linear-quadratic optimal control settings (for example Part III of Whittle (1982)), the certainty equivalence principle allows to decouple state estimation based on observations and control decisions. However, in more complicated settings such as what we consider here, certainty equivalence almost certainly doesn't hold.

In this paper, we augment the body of literature dealing with exploration vs. exploitation trade-offs in systems where a controller needs to choose a server (channel/resource/bandit) at any given time, and the choice influences both the immediate reward (service success) and the information obtained. A general class of such problems, denoted Reward Observing Restless Multi Armed Bandits (RORMAB), is outlined in Kuhn and Nazarathy (2015), where much previous literature is surveyed. Key contributions in this area are (Koole et al. 2001) and the more recent (Liu and Zhao 2010). The former finds the structure of optimal policies from first principles. The latter, generalizes the setting and utilizes the celebrated Whittle Index (Whittle 1988) for such a partially observable case. Related recent results dealing with RORMAB problems are in Larrañaga et al. (2016) and Larrañaga et al. (2014). Of further interest is the latest rigorous account on asymptotic optimality of the Whittle index, Verloop (2014).

Our focus in this paper is on presenting methodological means. With the aim of capturing the relationship between information and stability, we put forward the simplest non-trivial example that we could devise. It is a family of controlled queueing systems, where server environments vary and the controller (choosing servers) only observes partial information. The system has a single discrete time queue of jobs served by either Server 1 or Server 2 where each server environment is an independent two-state Markov chain also known as a Gilbert-Elliot channel (Sadeghi et al. 2008). A controller having (potentially) only partial information, selects one of the two servers at each time.

Application instances where service rates vary and are not fully known or observed are common. Examples include communication networks (where servers are communications channels), human service systems such as call-centers (where servers are employees) and energy systems where customer demand may represent the server. The simple model in this paper is designed to open the door for analyzing such systems and our focus is on the methodology.

The role of information is explored by considering different observation schemes. At one extreme, the controller has full information of the servers' environment states. At the other extreme, the controller is completely unaware of the servers' environment states. Obviously the stability region of the system in the latter situation is a subset of the former. Our contribution is in considering additional more realistic observation schemes. One such scheme is a situation where the controller only observes the environment state of the server currently chosen. This type of situation has been widely studied in some of the references mentioned above and surveyed in Kuhn and Nazarathy (2015), but most of the literature dealing with this situation does not consider stability of a queue. A more constrained scenario is one where the controller only observes the success/failure of service (from the server chosen) at every time slot. Such a partial observability situation was recently introduced in Nazarathy et al. (2015) in the context of stability and analysed in Meshram et al. (2016) with respect to the Whittle index. In Li and Neely (2013), stability of a related multi-server system was analysed.

An additional observation scheme that we consider is one where the server is only aware of the queue size process. In (non-degenerate) continuous time systems, such an observation scheme is identical to the former scheme. But an artefact of our discrete time model is that such a scheme reveals less information to the controller (this is due to the fact that both an arrival and a departure may occur simultaneously, going unnoticed by the controller).

With the introduction of the five observation schemes mentioned above, this work takes first steps towards analysing the effect of information on the achievable stability region. A controller of such a system makes use of a belief state implementation. We put forward (simple) explicit belief state update recursions for each of the observation schemes. These are then embedded in Bellman equations describing optimal solutions of associated Partially Observable Markov Decision Processes (POMDP). Numerical solution of the POMDPs then yields insight on structural properties and achievable stability regions. By construction, two-state Markov server environments are more predictable when the mixing times of the Markov chains increase. We quantify this use of channel-memory, through a combination of numerical and analytic results.

While it is often the case that MDPs (or POMDPs) associated with queueing models, can be cast as QBDs once a class of control policies is found, a general methodology for this is still lacking (see for example Mészáros and Telek (2014)). We advance this methodology in the current paper by presenting a detailed QBD model of the system. The virtue of our QBD based model is that we are able to quantify the effect of a finite state controller on the achievable stability region whose upper bound is given by an elegant matrix analytic expression. Our use of the QBD is for performance analysis of a given policy.

The remainder of this paper is as follows. In Section 2, we introduce the system model and observation schemes. In Section 3, we put forward recursions for belief state updates. In Section 4, we present the myopic policy and the Bellman equations for different observation schemes and present a numerical investigation. In Section 5, we construct a QBD representation of the system and find the stability criterion. We conclude in Section 6.

2 System Model

Our model consists of a discrete time queueing system with a single queue and two servers. However, at any time, only one of the two servers may be engaged. Such a situation may occur in practice when both servers make use of an additional shared resources. For example, the servers may represent different communication channels (frequencies) sharing the same antenna and cannot be used simultaneously. When presented with a job from the queue, the controller's choice is which server to assign to that job. While jobs are homogeneous, the server's capacity is dynamic via its random environment; where the state of the environment directly affects the rate (or quality) of service offered by the server. Hence, the control choice depends on the knowledge or belief of which server is "best" at any given time. In such a model, the quality of information affects system throughput and hence stability.

More specifically, consider a situation as depicted in Fig. 1. Jobs arrive into the queue, $Q(t)$, and are potentially served by one of two servers $j = 1$ or $j = 2$, according to some control policy. At most one of the two servers may operate at any given time. The system is operating in discrete time steps $t = 0, 1, \dots,$ where in each time step, the following sequence of events occur:

(1) An arrival occurs as indicated through $E(t)$: $E(t) = 1$ indicates an arrival and otherwise $E(t) = 0$. Note that for simplicity our model assumes at most one arrival per time slot. This is not an uncommon assumption (see for instance (Leng et al. 2016)).

(2) The environments of the servers update from $\big(X_1(t-1),\ X_2(t-1)\big)$ to $\big(X_1(t),\ X_2(t)\big)$, autonomously. That is, the updating of the environment is not influenced by arrivals, queue length and controller choice.

(3) A control decision $u = U(t)$ of which server to select is made based on observations in previous time steps, denoted by $Y(t-1), Y(t-2), \dots$. This is through a decision rule π. We consider different observation schemes as described below. Note that this description follows a control theory paradigm involving the environment state (X), observation (Y) and control choice (U). As our focus is on methodology for analyzing the role of information, we focus on different means for describing the observations (Y). For example, the "full observation" Y is simply the environment state X. Details follow.

(4) The control action is executed and the queue length is updated as follows:

$$Q(t+1) = Q(t) + E(t) - I(t), \qquad \text{with} \qquad I(t) = I^{(u)}_{X_u(t)}(t).$$

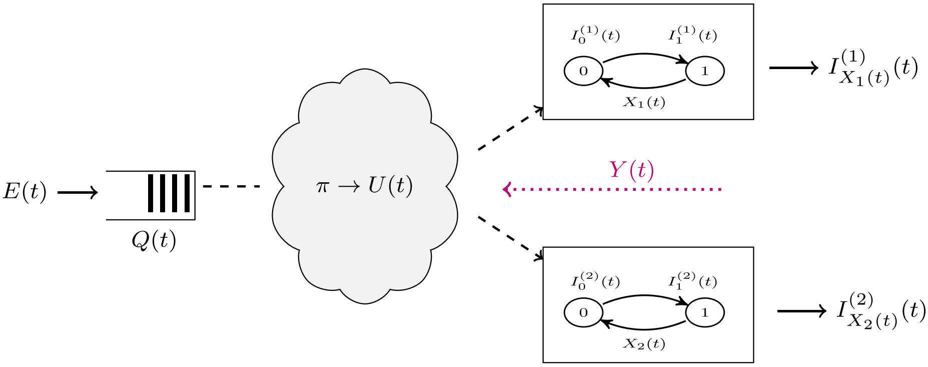

Fig. 1 A controller operating under a decision rule, π, decides at each time step, t, if to use server $U(t) = 1$ or $U(t) = 2$ based on previously observed information, $Y(t-1), Y(t-2), \cdots$. The server environment states, $X_i(t)$ are Markov modulated

Here, $I(t)$ is an indicator variable capturing whether there was a service success or not. It is constructed from the primitive sequences,

$$\{I_i^{(j)}(t),\ t = 0, 1, 2, \dots\},$$

for servers $j = 1, 2$ and server environment states $i = 0, 1$. Note that when $Q(t) = 0$, we notionally assume that $U(t) = u = 0$, indicating "no action" and in this case denote $I_i^{(0)} \equiv 0$ for $i = 0, 1$.

(5) The observation of $Y(t)$ is made and is used in subsequent time steps.

The sequence of events (1)–(5) as above repeats in every time step and fully defines the evolution law of the system.

The nature of the observations, Y, is described via the following distinct observation schemes:

(I) **Full observation**: The controller knows the environment state of both servers all the time. In this case

$$Y(t) = \big(X_1(t),\ X_2(t)\big),$$

and further, the sequence of steps above is slightly modified with step (5) taking place between steps (2) and (3) and the policy at step 3 being

$$u = U(t) = \pi\big(Y(t)\big).$$

(II) **State observation:** The controller observes the environment state of the selected server at time t, but does not observe the other server at that time. Hence

$$Y(t) = \begin{cases} \big(X_1(t), \emptyset\big) & \text{if } u = 1, \\ \big(\emptyset, X_2(t)\big) & \text{if } u = 2. \end{cases}$$

(III) **Output observation:** The controller observes the success or failure of outputs of the server selected (but gains no information about the other server at that time). Hence

$$Y(t) = \begin{cases} \big(I^{(1)}_{X_1(t)}(t), \emptyset\big) & \text{if } u = 1, \\ \big(\emptyset, I^{(2)}_{X_2(t)}(t)\big) & \text{if } u = 2. \end{cases}$$

(IV) **Queue observation:** The controller only observes the queue length, $Q(t)$, and can thus utilize the differences

$$\varDelta Q(t) = Q(t+1) - Q(t) = E(t) - I(t).$$

Note that since the system is operating in discrete time, there is some loss of information compared to case III : If $\varDelta Q(t) = 1$ or $\varDelta Q(t) = -1$, then it is clear that $I(t) = 0$ or $I(t) = 1$, respectively. But if $\varDelta Q(t) = 0$, then since the controller does not observe $E(t)$, there is not a definitive indication of $I(t)$.

(V) **No observation:** We assume the controller does not observe anything. Nonetheless, as with the other cases, the controller knows the system parameters as described below.

We consider the simplest non-trivial probably model for the primitives. These are $E(t)$, $I_i^{(j)}(t)$ and the environment processes, $X_j(t)$, all assumed mutually independent. Note that as common in queueing analysis, independent model primitives lead to highly non-trivial queueing models. For example, in queueing networks, it is common to assume that all primitive processes (arrivals, service times and routing decisions) are independent; still the resulting queueing networks are complicated models with intricate dependencies. See for example (Bramson 2008).

The arrivals, $E(t)$, are an i.i.d. sequence of Bernoulli random variables, each with probability of success λ. The service success indicators, for each server $j = 1, 2$ and environment state $i = 0, 1$, denoted by $\{I_i^{(j)}(t),\ t = 0, 1, 2, \ldots\}$, are each an i.i.d. sequence with

$$I_i^{(j)}(t) \sim \text{Bernoulli}\big(\mu_i^{(j)}\big).$$

Moreover, for setting up a non-trivial situation and without loss of generality, we assume

$$\mu_0^{(2)} \le \mu_0^{(1)} < \mu_1^{(1)} \le \mu_1^{(2)}. \tag{1}$$

Hence environment states $i = 1$ for both servers are better than environment states $i = 0$ (this is without loss of generality). Further, the spread of the chance of success for Server 2 is greater or equal to that for Server 1. Note that if the both environment states of one server would be superior to the environment states of another server, the problem would be trivial as the controller would always use the superior server. See Fig. 2.

For the state environment processes, we restrict attention to a two-state Markov chain, sometimes referred to as a Gilbert–Elliot channel. We denote the probability transition matrix for environment state (referred to as "state" in the sequel) of server j as:

$$P^{(j)} = \begin{bmatrix} \bar{p}_j & p_j \\ q_j & \bar{q}_j \end{bmatrix} = \begin{bmatrix} 1 - \gamma_j\,\bar{\rho}_j & \gamma_j\,\bar{\rho}_j \\ \bar{\gamma}_j\,\bar{\rho}_j & 1 - \bar{\gamma}_j\,\bar{\rho}_j \end{bmatrix}, \tag{2}$$

with $\bar{x} := 1 - x$. In the sequel, we omit the server index j from the individual parameters of $P^{(j)}$. A standard parametrization of this Markov chain uses transition probabilities $p, q \in [0, 1]$. Alternatively, we may specify the stationary probability of being in state 1, denoted by $\gamma \in [0, 1]$, together with the second eigenvalue of P, denoted by $\rho \in \big[1 - \min(\gamma^{-1}, \bar{\gamma}^{-1}),\ 1\big]$.

Using this parametrization, ρ quantifies the time-dependence of the chain; when $\rho = 0$ the chain is i.i.d., otherwise there is memory. If $\rho > 0$ then environment states are positively correlated, otherwise they are negatively correlated. Our numerical examples in this paper deal with positive correlation as it is often the more reasonable model for channel memory. Positive correlation captures channel memory while ensuring that environments do not alternate in a haphazard manner. As opposed to that, the negatively correlated case appears to be a mathematical artefact of the two-state Markov chain in discrete time and is a less realistic assumption. See Kuhn and Nazarathy (2015) for further discussion.

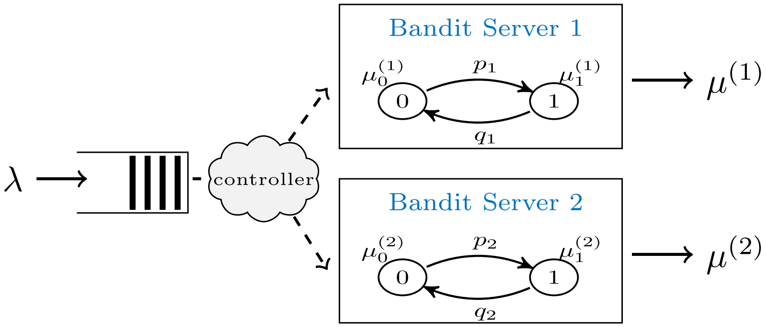

Fig. 2 Parameters of the system

The relationship between the (γ, ρ) parametrization and the (p, q) parametrization is given by $p = \gamma\,\bar{\rho}, q = \bar{\gamma}\,\bar{\rho}, \gamma = p/(p+q)$, and $\rho = 1 - p - q$.

It is instructive to consider the long term behaviour of $I^{(j)}_{X_j(t)}(t)$ for $j = 1, 2$ by assuming the sequence $\{X_j(t),\ t = 0, 1, 2, \dots\}$ is stationary and thus each $X_j(t)$ is Bernoulli distributed with parameter γ_j. In this case

$$\begin{aligned}\mathbb{E}\left[I^{(j)}_{X_j(t)}(t)\right] &= \bar{\gamma}_j\,\mu^{(j)}_0 + \gamma_j\,\mu^{(j)}_1\,,\\ \mathrm{Var}\left[I^{(j)}_{X_j(t)}(t)\right] &= \bar{\gamma}_j\,\mu^{(j)}_0\,\bar{\mu}^{(j)}_0 + \gamma_j\,\mu^{(j)}_1\,\bar{\mu}^{(j)}_1 + \gamma_j\,\bar{\gamma}_j\left(\mu^{(j)}_1 - \mu^{(j)}_0\right)^2.\end{aligned}$$

These quantities are useful for obtaining a rough handle on the performance of the system.

Stability analysis of queueing systems has received much attention in the past few decades, see Bramson (2008). In more classic queues, stability is straightforward as it is immediately evident by comparing the arrival rate and the service rate (consider for example the M/M/1 queue). However, in complicated systems, stability is a pressing performance analysis question that often requires non-trivial analysis. The exact definition of stability often varies based on the mathematical model used. In our methodology, we consider the *stability region* of the system to be the set of arrival rates for which there exists a decision rule, π, under which the Markov chain describing the system is positive recurrent. A technical point is that for the same queueing model, there may be different Markov chain representations, and hence positive recurrence depends on the representation used. One concrete description is in Section 5 where we make use of stability results from Matrix Analytic Methods.

In highly complex queueing models, one may observe non-monotone stability regions (see for example Bramson (2008) for some obscure examples). However, in most simple models, when considering the stability region in terms of the arrival rate λ, it holds that the stability region is of the form

$$\{\lambda\ :\ \lambda < \mu^*\}.$$

Our analysis for our model indicates that this is the case, where the *stability bound* μ^* varies according to the observation schemes I–V above and is identical to the maximal throughput rate that may be obtained in a system without a queue (but rather with an infinite supply of jobs).

Our methodological contribution is to quantify how the stability bound, μ^*, depends on the observation scheme. The construction of μ^* depends on the optimal policy that can be used with the given information. In this respect, one may view “lack of information” as a constraint on the optimal policy. For example, an “unconstrained” case is that with full information and a “fully constrained” case is with no information. This argumentation implies that the more information that the controller has, the less constraints and thus the higher the stability bound. Hence,

$$\mu^*_{\text{no}} \le \mu^*_{\text{queue}} \le \mu^*_{\text{output}} \le \mu^*_{\text{state}} \le \mu^*_{\text{full}}\,, \tag{3}$$

where μ^*_{no} corresponds to case V, μ^*_{queue} corresponds to case IV and so forth.

The lower and upper bounds, μ^*_{no} and μ^*_{full}, are easily obtained as we describe now. However, the other cases are more complicated and are analysed in the sections that follow. For the lower and upper bounds we have

$$\begin{aligned}\mu^*_{\text{no}} &= \max\left\{\mathbb{E}\left[I^{(1)}_{X_1(t)}(t)\right],\ \mathbb{E}\left[I^{(2)}_{X_2(t)}(t)\right]\right\},\\ \mu^*_{\text{full}} &= \bar{\gamma}_1\bar{\gamma}_2\mu^{(1)}_0 + \gamma_2\mu^{(2)}_1 + \gamma_1\bar{\gamma}_2\mu^{(1)}_1.\end{aligned}$$

The lower bound, μ^*_{no} is trivially achieved with a control policy that always uses the server with the higher mean throughput. The upper bound, is achieved with a control policy that uses the best server at any given time. Under the ordering in (1), the throughput in this case is calculated as follows: If both servers are in state 0, then since $\mu_0^{(2)} \le \mu_0^{(1)}$, the controller selects Server 1. This situation occurs at a long term proportion, $\bar{\gamma}_1\bar{\gamma}_2$, hence we obtain the first term of μ^*_{full}. The other terms of μ^*_{full} are obtained with a similar argument. Note that when $\gamma_1 = \gamma_2 = \gamma$ and $\mu_i^{(1)} = \mu_i^{(2)} = \mu_i$ for $i = 0, 1$, the expression is reduced to $\mu^*_{\text{full}} = \overline{\gamma}^2\mu_0 + (1 - \overline{\gamma}^2)\mu_1$ and can be obtained by a Binomial argument.

As a benchmark numerical case, all the examples we present use

$$\gamma = 0.5\,,\ \ \mu_0 = 0.2\,,\ \ \mu_1 = 0.8\,, \tag{4}$$

for both servers. Under these parameters

$$\mu^*_{\text{no}} = 0.5\,, \qquad \text{and} \qquad \mu^*_{\text{full}} = 0.65\,.$$

Hence in the examples that follow, we explore how μ^*_{queue}, μ^*_{output} and μ^*_{state} vary within the interval $[0.5, 0.65]$ as ρ_j, $j = 1, 2$ varies.

3 Belief States

In implementing a controller for each of the observation schemes, the use of *belief states* reduces both the complexity of the controller and the related analysis. The notion of belief states is an integral part of POMDP (see Smallwood and Sondik 1973). The idea is to summarize the history of observations, $Y(t-1), Y(t-2), \ldots,$ via a probability distribution over the state space. In our case, we are able to go further by describing the probability distribution via *sufficient statistics* that are updated by the controller. Since the state of each server takes one of two possible values, a natural choice for the belief state of server j is

$$\omega_j(t) = \mathbb{P}\big(X_j(t) = 1 \mid \text{Prior knowledge to time } t\big).$$

As we describe now, it is a simple matter to recursively update this sequence in a Bayesian manner. Denoting $\omega_j(t)$ by ω, the believed chance of success is

$$r(\omega) := \bar{\omega}\mu_0 + \omega\mu_1.$$

The updating algorithms (different for each observation scheme) make use of the following:

$$\begin{aligned} \tau_n(\omega) &:= \bar{\omega}\gamma\bar{\rho} + \omega(1 - \bar{\gamma}\bar{\rho}) = \omega\rho + \gamma\bar{\rho}\,, \qquad \tau_f(\omega) := \tfrac{\bar{q}\,\bar{\mu}_1\,\omega + p\,\bar{\mu}_0\,\bar{\omega}}{\bar{r}(\omega)}\,, \\ \tau_s(\omega) &:= \tfrac{\bar{q}\,\mu_1\,\omega + p\,\mu_0\,\bar{\omega}}{r(\omega)}, \quad \tau_c(\omega) := \lambda\tau_s(\omega) + \bar{\lambda}\tau_f(\omega). \end{aligned} \tag{5}$$

Note that in the above, superscripts j are omitted for clarity. The probabilistic meaning of these functions is described in the sequel. These are used to define recursions for updating the belief state. Each observation scheme entails a different type of recursion:

(II) State observation:

$$\big(\omega_1(t+1), \omega_2(t+1)\big) = \begin{cases} \big(X_1(t), \tau_n^{(2)}\big(\omega_2(t)\big)\big), & U(t) = 1, \\ \big(\tau_n^{(1)}\big(\omega_1(t)\big), X_2(t)\big), & U(t) = 2. \end{cases}$$

(III) Output observation:

$$
\big(\omega_1(t{+}1),\omega_2(t{+}1)\big)=\begin{cases}
\big(\tau_f^{(1)}(\omega_1(t)),\tau_n^{(2)}\big(\omega_2(t)\big)\big), & I^{(1)}_{X_1(t)}(t)=0, \\
 & \qquad\qquad U(t)=1,\\
\big(\tau_s^{(1)}(\omega_1(t)),\tau_n^{(2)}\big(\omega_2(t)\big)\big), & I^{(1)}_{X_1(t)}(t)=1,\\
\\
\big(\tau_n^{(1)}\big(\omega_1(t)\big),\tau_f^{(2)}(\omega_2(t))\big), & I^{(2)}_{X_2(t)}(t)=0,\\
 & \qquad\qquad U(t)=2.\\
\big(\tau_n^{(1)}\big(\omega_1(t)\big),\tau_s^{(2)}(\omega_2(t))\big), & I^{(2)}_{X_2(t)}(t)=1,
\end{cases}
$$

(IV) Queue observation:

$$
\big(\omega_1(t{+}1),\omega_2(t{+}1)\big)=\begin{cases}
\big(\tau_f^{(1)}(\omega_1(t)),\tau_n^{(2)}\big(\omega_2(t)\big)\big), & \Delta Q(t)=1,\\
\big(\tau_c^{(1)}(\omega_1(t)),\tau_n^{(2)}\big(\omega_2(t)\big)\big), & \Delta Q(t)=0, \qquad U(t)=1,\\
\big(\tau_s^{(1)}(\omega_1(t)),\tau_n^{(2)}\big(\omega_2(t)\big)\big), & \Delta Q(t)=-1,\\
\\
\big(\tau_n^{(1)}\big(\omega_1(t)\big),\tau_f^{(2)}(\omega_2(t))\big), & \Delta Q(t)=1,\\
\big(\tau_n^{(1)}\big(\omega_1(t)\big),\tau_c^{(2)}(\omega_2(t))\big), & \Delta Q(t)=0, \qquad U(t)=2.\\
\big(\tau_n^{(1)}\big(\omega_1(t)\big),\tau_s^{(2)}(\omega_2(t))\big), & \Delta Q(t)=-1,
\end{cases}
$$

Upon applying the recursions above, we indeed track the belief state as needed.

Proposition 1 *For each of the observation schemes, assume that at $t=0$, $\omega_j(0)=\mathbb{P}\big(X_j(0)=1\big)$. Then upon implementing the recursion above,*

$$\omega_j(t)=\mathbb{P}\big(X_j(t)=1\,|Y(t),Y(t-1),\ldots,Y(0)\big),\qquad t=1,2,\ldots.$$

Proof The proof and derivation of the operators in (5) follows from elementary conditional probabilities and induction. We illustrate this for the output observation case here. It holds that

$$
\begin{aligned}
\mathbb{P}\Big(X(t)=1\;\Big|\;I(t-1)=0\Big) &= \frac{\mathbb{P}\left(X(t)=1,I(t-1)=0\right)}{\mathbb{P}\left(I(t-1)=0\right)}\\
=\ &\frac{\mathbb{P}\left(X(t)=1,I=0\,\middle|\,X=1\right)\mathbb{P}\left(X=1\right)+\mathbb{P}\left(X(t)=1,I=0\,\middle|\,X=0\right)\mathbb{P}\left(X=0\right)}{\mathbb{P}\left(I=0\,\middle|\,X=1\right)\mathbb{P}\left(X=1\right)+\mathbb{P}\left(I=0\,\middle|\,X=0\right)\mathbb{P}\left(X=0\right)},
\end{aligned}
$$

where we denote $X=X(t-1)$ and $I=I(t-1)$. Since $X(t)$ and $I(t-1)$ are conditionally independent given $X(t-1)$, the above numerator can be written as:

$$
\begin{aligned}
&\mathbb{P}\left(X(t)=1\,\middle|\,X=1\right)\mathbb{P}\left(I=0\,\middle|\,X=1\right)\mathbb{P}\left(X=1\right)\\
+\ &\mathbb{P}\left(X(t)=1\,\middle|\,X=0\right)\mathbb{P}\left(I=0\,\middle|\,X=0\right)\mathbb{P}\left(X=0\right)=\bar{q}\,\bar{\mu}_1\,\omega+p\,\bar{\mu}_0\,\bar{\omega}.
\end{aligned}
$$

Similarly, for the denominator we have:

$$\mathbb{P}\left(I=0\,\middle|\,X=1\right)\mathbb{P}\left(X=1\right)+\mathbb{P}\left(I=0\,\middle|\,X=0\right)\mathbb{P}\left(X=0\right)=\bar{\mu}_1\omega+\bar{\mu}_0\bar{\omega},$$

which is equal to $\bar{r}(\omega)$. Hence as expected, we find that

$$\mathbb{P}\Big(X(t)=1\;\Big|\;I(t-1)=0\Big)=\tau_f(\omega).$$

The derivation of $\tau_n(\omega)$ and $\tau_s(\omega)$ follows similar lines. For the queue observation case, notice that

$$\mathbb{P}\Big(X(t)=1 \mid \varDelta Q(t)=1\Big) = \tau_f(\omega)\,, \qquad \mathbb{P}\Big(X(t)=1 \mid \varDelta Q(t)=-1\Big) = \tau_s(\omega)\,,$$
$$\mathbb{P}\Big(X(t)=1 \mid \varDelta Q(t)=0\Big) = \lambda\tau_s(\omega)+\bar{\lambda}\tau_f(\omega)=\tau_c(\omega).$$

The state observation case follows similar lines. □

Note that the fixed point of τ_n is the stationary probability γ. The fixed points of τ_f and τ_s are also of interest. When $\rho \neq 0$ and $\mu_0 \neq \mu_1$, τ_f and τ_s are (real) hyperbolic Möbius transformations of the form $(a\omega+b)/(c\omega+d)$ for $\omega \in [0,1]$. As such, they each have two distinct fixed points, one stable and one unstable. Here, excluding trivialities where $p, q \in \{0,1\}$, the stable fixed point of each lies in $(0,1)$ and is of the form $\big(a-d+\sqrt{(a-d)^2+4bc}\big)/2c$ (see also Lemma 2 and 3 of MacPhee and Jordan (1995)). For τ_f, we have $a=\bar{q}\,\bar{\mu}_1 - p\,\bar{\mu}_0$, $b=p\,\bar{\mu}_0$, $c=\bar{\mu}_1-\bar{\mu}_0$, and $d=\bar{\mu}_0$. Fixed point of τ_s comes from the same formula by replacing $\bar{\mu}_i$ by μ_i.

Denote by $\omega_s^{(j)}$ and $\omega_f^{(j)}$, the stable fixed point of $\tau_s^{(j)}$ and $\tau_f^{(j)}$, respectively for $j=1,2$. Then

$$\varOmega = \varOmega_1 \times \varOmega_2 \subset [0,1]\times[0,1],$$

where we put

$$\varOmega_j = [\min(\omega_f^{(j)}, \omega_s^{(j)}),\ \max(\omega_f^{(j)}, \omega_s^{(j)})],$$

is the *belief state space* and the limit of any infinite subsequence of the mappings τ_n, τ_f, τ_s and τ_c (for $\omega_1, \omega_2 \in [0,1]$) lies within $\varOmega$, see Nazarathy et al. (2015) for more details.

4 Maximal Throughput

Having defined sufficient statistics for the belief state and their evolution, the problem of finding a maximally stabilizing control can be posed as a Partially Observable Markov Decision Process (POMDP), see for example Bäuerle and Rieder (2011) or the historical reference (Smallwood and Sondik 1973). The objective for the POMDP is

$$\mu^* = \sup_{\pi} \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi\left[\sum_{t=0}^{T-1} I(t)\right],$$

where $U(t)=\pi\big(\omega_1(t),\omega_2(t)\big)$ influences $I(t)$ as outlined in Section 2. A formal treatment of the POMDP, relating it to the maximal stability region of the system can be carried out. We now first introduce the myopic policy for the POMDP and then move onto optimality equations.

The Myopic Policy One specific policy is the *myopic policy* given by:

$$\pi(\omega_1,\omega_2) = \begin{cases} \text{Server 2 if} & \omega_2 \geq \frac{\mu_1^{(1)}-\mu_0^{(1)}}{\mu_1^{(2)}-\mu_0^{(2)}}\,\omega_1 + \frac{\mu_0^{(1)}-\mu_0^{(2)}}{\mu_1^{(2)}-\mu_0^{(2)}}, \\ \text{Server 1 if} & \text{otherwise.} \end{cases} \tag{6}$$

The affine threshold in this policy is obtained by comparing the immediate expected mean throughput for any given pair (ω_1,ω_2) and choosing the server that maximizes it. Such a policy is attractive in that it is easy to implement. Further, when the servers are symmetric

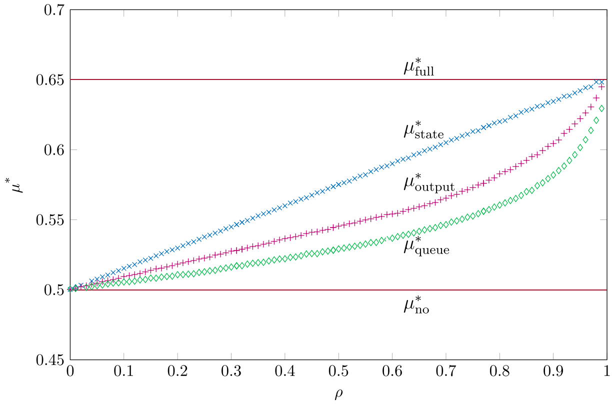

Fig. 3 The stability bound is displayed as a function of ρ for the various observation schemes. This plot is based on simulation results using the parameters in (4) and $\rho_1 = \rho_2 = \rho$

(all parameters are identical), it holds from symmetry that it is optimal. In this case it can be represented as

$$\pi(\omega_1, \omega_2) = \text{argmax}_{i=1,2}\, \omega_i,$$

and we refer to it as the *symmetric myopic policy*.

Simulation Result Figure 3 demonstrates results obtained through a Monte Carlo simulation[1] of the model for observation schemes (I)–(V). We use the parameters in (4) and vary ρ with $\rho_1 = \rho_2 = \rho$ in the range [0, 1) with steps of 0.01. The policy used is the symmetric myopic policy and it is optimal since the servers are identical.

As we see from the figure, the ordering (3) appears to hold. Further, at the i.i.d. case $\rho = 0$, historical observations are not useful and the throughput of all observations schemes, except for full observation, is at 0.5. At the other extreme, when $\rho \to 1$, we have that the state observation scheme converges to a throughput identical to that of the full observation scheme. This is because at that regime, the server states rarely change. Thus, from a throughput perspective, the controller behaves as though it has full information. On the other hand, even at $\rho = 1$, the output observation scheme and queue observation scheme still perform at a lower throughput. Finally, for $\rho \in (0, 1)$ it is evident that there is a gap in performance for each observation scheme. This gap quantifies the value of information in controlling our model and motivates further analysis.

Bellman Equations It is well-known that the optimal policy that maximizes the throughput follows from the average reward Bellman equations. See Puterman (2014) for background on Markov Decision Processes (MDP) or Hernández-Lerma and Lasserre (2012) for a discussion of average reward optimality with such state spaces. The Bellman equation is then

$$\mu^* + h(\omega_1, \omega_2) = \max\left\{h^{(1)}(\omega_1, \omega_2),\ h^{(2)}(\omega_1, \omega_2)\right\},$$

where h is the relative value function and the individual components $h^{(j)}(\cdot, \cdot)$, vary as follows:

[1] Simulation details: We run the process for $t = 5,000,000$ time units, using common random numbers for each run and recording the average throughput.

(II) State observation:

$$\begin{aligned} h^{(1)}(\omega_1,\omega_2) &:= r^{(1)}(\omega_1) + \Big[\bar{\omega}_1\, h\big(p_1, \tau_n^{(2)}(\omega_2)\big) + \omega_1\, h\big(\bar{q}_1, \tau_n^{(2)}(\omega_2)\big)\Big], \\ h^{(2)}(\omega_1,\omega_2) &:= r^{(2)}(\omega_2) + \Big[\bar{\omega}_2\, h\big(\tau_n^{(1)}(\omega_1), p_2\big) + \omega_2\, h\big(\tau_n^{(1)}(\omega_1), \bar{q}_2\big)\Big]. \end{aligned}$$

(III) Output observation:

$$\begin{aligned} h^{(1)}(\omega_1,\omega_2) &:= r^{(1)}(\omega_1) + \Big[\bar{r}^{(1)}(\omega_1)\, h\big(\tau_f^{(1)}(\omega_1), \tau_n^{(2)}(\omega_2)\big) \\ &\qquad + r^{(1)}(\omega_1)\, h\big(\tau_s^{(1)}(\omega_1), \tau_n^{(2)}(\omega_2)\big)\Big], \\ h^{(2)}(\omega_1,\omega_2) &:= r^{(2)}(\omega_2) + \Big[\bar{r}^{(2)}(\omega_2)\, h\big(\tau_n^{(1)}(\omega_1), \tau_f^{(2)}(\omega_2)\big) \\ &\qquad + r^{(2)}(\omega_2)\, h\big(\tau_n^{(1)}(\omega_1), \tau_s^{(2)}(\omega_2)\big)\Big]. \end{aligned}$$

(IV) Queue observation:

$$\begin{aligned} h^{(1)}(\omega_1,\omega_2) := r^{(1)}(\omega_1) &+ \Big[\lambda \bar{r}^{(1)}(\omega_1)\, h\big(\tau_f^{(1)}(\omega_1), \tau_n^{(2)}(\omega_2)\big) \\ &\quad + \bar{\lambda} r^{(1)}(\omega_1)\, h\big(\tau_s^{(1)}(\omega_1), \tau_n^{(2)}(\omega_2)\big) \\ &+ \Big(\bar{\lambda}\bar{r}^{(1)}(\omega_1) + \lambda r^{(1)}(\omega_1)\Big)\Big(h(\tau_c^{(1)}(\omega_1), \tau_n^{(2)}(\omega_2))\Big)\Big], \\ h^{(2)}(\omega_1,\omega_2) := r^{(2)}(\omega_2) &+ \Big[\lambda \bar{r}^{(2)}(\omega_2)\, h\big(\tau_n^{(1)}(\omega_1), \tau_f^{(2)}(\omega_2)\big) \\ &\quad + \bar{\lambda} r^{(2)}(\omega_2)\, h\big(\tau_n^{(1)}(\omega_1), \tau_s^{(2)}(\omega_2)\big) \\ &+ \Big(\bar{\lambda}\bar{r}^{(1)}(\omega_2) + \lambda r^{(1)}(\omega_2)\Big)\Big(h(\tau_n^{(1)}(\omega_1), \tau_c^{(2)}(\omega_2))\Big)\Big]. \end{aligned}$$

The optimal decision is then to choose Server 1 if and only if $h^{(1)}(\omega_1,\omega_2) \geq h^{(2)}(\omega_1,\omega_2)$, breaking ties arbitrarily. Since $\tau_n^{(j)}(0) = \tau_s^{(j)}(0) = \tau_f^{(j)}(0) = p_j$ and $\tau_n^{(j)}(1) = \tau_s^{(j)}(1) = \tau_f^{(j)}(1) = \bar{q}_j$, for all three aforementioned cases we have:

$$\begin{aligned} \mu^* + h(0,0) &= \max\big\{\mu_0^{(1)} + h(p_1,p_2), \mu_0^{(2)} + h(p_1,p_2)\big\}, \\ \mu^* + h(1,1) &= \max\big\{\mu_1^{(1)} + h(\bar{q}_1,\bar{q}_2), \mu_1^{(2)} + h(\bar{q}_1,\bar{q}_2)\big\}. \end{aligned} \tag{7}$$

From the ordering in (1), the above equations imply that at point $(\omega_1,\omega_2) = (0,0)$, choosing Server 1 is optimal and at point $(\omega_1,\omega_2) = (1,1)$ choosing Server 2 is optimal. This observation gives some initial insight into the structure of optimal policies. We now pursue these further numerically.

Numerical Investigation of Optimal Policies A solution to the above Bellman equations can be obtained numerically using relative value iteration and discretization of the belief state space, Ω. Here, we applied the MDP toolbox in Matlab (see Chadés et al. 2014) by considering that each interval [0, 1] for ω_1 and ω_2 is partitioned to 1000 equal sub-intervals. We ran relative value iteration with an accepted error set to 10^{-4}. Our various numerical experiments indicate the following:

1. The ordering in (3) holds.
2. Increasing (positive) ρ_j always yields an increase in μ^*.
3. Though the myopic policy does not appear to be generally optimal, when both servers are identical, the optimal policy is the symmetric myopic policy.

4. In all cases, the optimal policy is given by a non-decreasing switching curve within Ω. That is, there exists a function $\omega_2^*(\omega_1)$ where the optimal policy is

$$\pi(\omega_1,\omega_2) = \begin{cases} \text{Server 2 if} & \omega_2 \geq \omega_2^*(\omega_1), \\ \text{Server 1 if} & \text{otherwise.} \end{cases}$$

5. When the ordering in (1) has strict inequalities, $\omega_2^*(0) > 0$ and $\omega_2^*(1) < 1$.
6. For identical servers, it holds that the switching curve for the output observation case is sandwiched between the switching curve of the state observation case and the myopic switching line (6).
7. The switching curve for the queue observation case depends on λ. Further, when λ is at either of the extreme points ($\lambda = 0$ or $\lambda = 1$), the queue observation case agrees with the output observation case.
8. Consider the spread of μ as in Asanjarani (2016) and denote the spread of μ for each server by ε_j where $\mu_0^{(j)} = \gamma_j - \varepsilon_j$ and $\mu_1^{(j)} = \gamma_j + \varepsilon_j$. Then, increasing spread of servers, ε_j, in all observation schemes II, III, and IV, yields an increase in μ^*.

As an illustration of some of the above properties, consider Table 1 based on the parameters of (4) and various values of ρ_1 and ρ_2. The results in the table further affirm comments 1 and 2 above and contains values that agree with Monte Carlo simulation results, similar to those of Fig. 3.

Figure 4 demonstrates the switching curves for output observation case III derived by considering $\varepsilon_1 = 0.3$, $\gamma = \rho = 0.5$ for both servers, and increasing the value of ε_2. The other cases' figures are similar. In this figure, Server 2 is called a *safe server* where the service success through both states of Server 2 are the same, or $\varepsilon_2 = 0$. Figure 4 also confirms comment 3 above.

As a further illustration, Fig. 5 presents switching curves, $w_2^*(\cdot)$ for the parameters of (4) with $\rho_1 = 0.5$ and $\rho_2 = 0.7$. In the figure, the red dotted line is the myopic policy line (suboptimal). The blue solid curve is the switching curve for the output observation case. The green dash dotted line is related to the queue observation case and the orange densely dashed curve is the switching curve for the state observation case. These curves were obtained by finding the optimal decision for every (discretized) element of Ω.

Figure 6 presents switching curves of queue observation scheme (case IV) for different values of λ with given parameters in (4). Moreover, we consider the changes in value of μ^* for different values of λ.

5 QBD Structured Models for Finite State Controllers

Here we illustrate how Matrix-Analytic Modelling (MAM) can be used to evaluate a finite state controller that approximates an optimal controller once such a controller is specified

Table 1 Stability region bounds for observation schemes (II)-(IV) for various ρ_1 and ρ_2 values

ρ_1	ρ_2	μ^*_{state}	μ^*_{output}	μ^*_{queue}
0.2	0.5	0.5543	0.5314	0.5190
0.4	0.5	0.5673	0.5400	0.5231
0.6	0.5	0.5823	0.5489	0.5289
0.8	0.5	0.6009	0.5647	0.5360

The queue observation case is with $\lambda = 0.5$

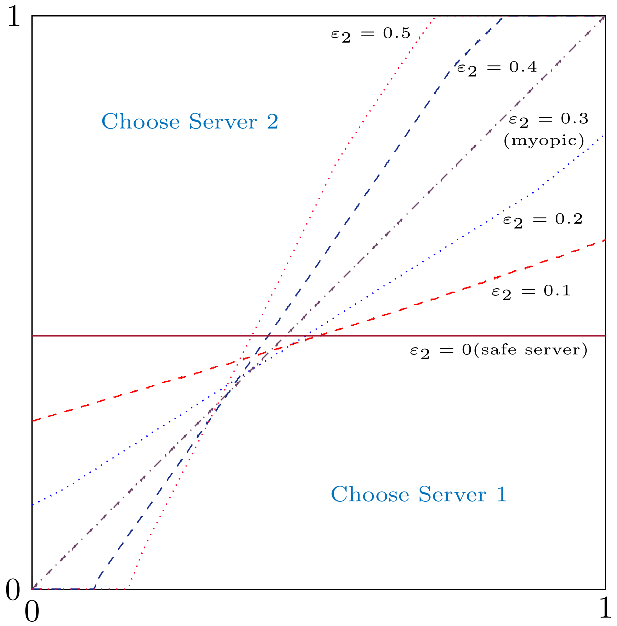

Fig. 4 Output observation (case III) switching curves for increasing spread

(e.g. by solving the Bellman equations above). Our analysis is for the output observation scheme (Case III). Similar analysis can be applied to the other observation schemes as well.

A finite state controller operates by using a finite discrete belief state $\tilde{\Omega}$, representing a discrete grid in Ω. With such a controller, we consider the whole system as a Quasi-Birth-and-Death (QBD) process (for more details about QBD process see for example Latouche and Ramaswami (1999)). Using the QBD structure, we find a matrix analytic expression for

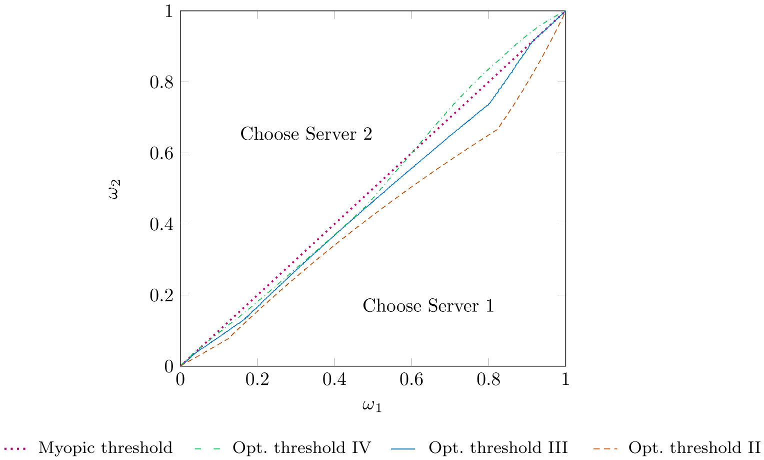

Fig. 5 Myopic and optimal policies for the state observation Case II, output observation Case III and queue observation Case IV ($\lambda = 0.5$). This is for a system with $\rho_1 = 0.5$ and $\rho_2 = 0.7$

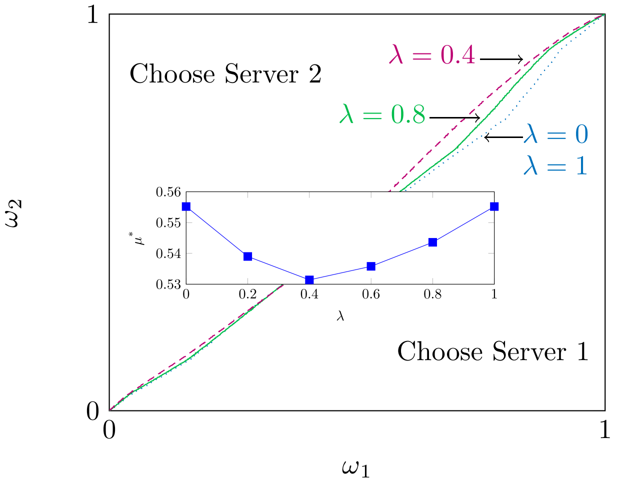

Fig. 6 Queue observation (case IV) switching curves and changes in value of μ^* for different values of λ by considering the other parameters as in (4)

μ^*_{output} (denoted by μ^* in this section). A further virtue of QBD modelling is that we can easily use the QBD framework for performance analysis of queue lengths distributions and related measures. Note that if we were to enhance the model to allow mulitple arrivals per time slot, then similar analysis could be carried out using G/M/1 type QBDs.

Take $\tilde{\Omega} = \{1, \ldots, M\}^2$ and define the controller state at time t by $(\psi_1(t), \psi_2(t)) \in \tilde{\Omega}$. In doing so, we treat $\psi_j(t)$ as $\lceil M\omega_j(t) \rceil$. The controller action is (potentially) randomized based on a matrix of probabilities C so that Server 2 is chosen with probability $C_{(\psi_1(t), \psi_2(t))}$ and otherwise the choice is Server 1. That is, the matrix C is a randomized control policy. Such a policy encodes information as in Fig. 5.

The controller state is updated in a (potentially) randomized manner based on the $M \times M$ stochastic matrices $N^{(j)}$, $S^{(j)}$, $F^{(j)}$ for $j = 1, 2$. The rows of matrices $N^{(j)}$, $S^{(j)}$ and $F^{(j)}$ indicate how to (potentially randomly) choose the next controller state. Here, N stands for No service, S for Success, and F for Failure as follows: if Server 1 was not selected (no service either because there were no jobs in the queue, or because the other server was selected), the distribution of the new state is $\left(N^{(1)}_{\psi_1(t),1}, \ldots, N^{(1)}_{\psi_1(t),M}\right)$; that is taken from the row indexed by $\psi_1(t)$. Similarly, if Server 1 was chosen and service was successful ($I = 1$), the distribution of the new state is $\left(S^{(1)}_{\psi_1(t),1}, \ldots, S^{(1)}_{\psi_1(t),M}\right)$. Finally, if Server 1 was chosen and the service failed ($I = 0$), the distribution of the new state is $\left(F^{(1)}_{\psi_1(t),1}, \ldots, F^{(1)}_{\psi_1(t),M}\right)$. Similarly, for Server 2, we have $\left(N^{(2)}_{1,\psi_2(t)}, \ldots, N^{(2)}_{M,\psi_2(t)}\right)$, $\left(S^{(2)}_{1,\psi_2(t)}, \ldots, S^{(2)}_{M,\psi_2(t)}\right)$ and $\left(F^{(2)}_{1,\psi_2(t)}, \ldots, F^{(2)}_{M,\psi_2(t)}\right)$, respectively.

We construct the matrices $N^{(j)}$, $S^{(j)}$, $F^{(j)}$ based on a discretization of $\tau_n^{(j)}$, $\tau_s^{(j)}$ and $\tau_f^{(j)}$, respectively. For example, construction of S from τ_s is as follows: the row elements of S by putting $j = M\tau_s\left(\dfrac{i-1}{M}\right)$ for each $i = 1, \cdots, M$ are given by

$$\begin{cases} S_{i,j} = 1 & j \text{ is an integer}, \\ S_{i,\lfloor j \rfloor} = 1 & 1 \leq \lfloor j \rfloor \leq M, \\ S_{i,\lceil j \rceil} = 1 & \text{otherwise}, \end{cases}$$

and $S_{i,k} = 0$ for all other row elements with $k \neq j$ and $k = 1, \dots, M$. After this, we ensure irreducibility of this matrix by fixing $\epsilon > 0$ (for instance, equal to 0.001 as in our numerical examples) and adding ϵ/M to each element of the matrix and then renormalizing it.

The matrices F and N are constructed in a similar way based on τ_f and τ_n, respectively. This is simply a mechanism to encode the transition operators over the finite grid. Hence the matrices $N^{(j)}$, $S^{(j)}$, $F^{(j)}$ describe propagation of ψ_j through the belief operators, similarly to the propagation of ω through their continuous counterparts.

Now, given such a controller with

$$\text{Controller parameters} = \Big(N^{(1)}, S^{(1)}, F^{(1)}, N^{(2)}, S^{(2)}, F^{(2)}, C\Big),$$

we construct a Markov chain, $Z(t)$ for the system. The state of this model at time t is given by the queue length, server environment state, and controller state as follows:

$$Z(t) = \Big(\underbrace{Q(t)}_{\text{Level}}, \underbrace{\Big(\overbrace{\big(X_1(t), X_2(t)\big)}^{\text{Servers}}, \overbrace{\big(\psi_1(t), \psi_2(t)\big)}^{\text{Controller}}\Big)}_{\text{Phase}}\Big) \in \{0, 1, \dots\} \times \{1, 2\}^2 \times \{1, \dots, M\}^2.$$

Explicit QBD Construction When the states of $Z(t)$ are lexicographically ordered, with first component countably infinite (levels) and the other components finite (phases), the resulting (infinite) probability transition matrix is of the QBD form:

$$A = \begin{bmatrix} \tilde{A}_0 & \tilde{A}_1 & & & & 0 \\ A_{-1} & A_0 & A_1 & & & \\ & A_{-1} & A_0 & A_1 & & \\ & & A_{-1} & A_0 & A_1 & \\ 0 & & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{8}$$

where each of $\tilde{A}_0$, $\tilde{A}_1$, A_{-1}, A_0, and A_1 is a block matrix of order $4M^2$ as we construct below.

The matrix A_{-1} represents the phase transition where there is a one level decrease. Similarly, the matrix A_1 represents phase transition where there is a one level increase and A_0 represents the phase transition where the level remains the same. The blocks are constructed as follows:

$$\tilde{A}_0 = \overline{\lambda}\tilde{N}, \quad \tilde{A}_1 = \lambda\tilde{N}, \quad A_{-1} = \overline{\lambda}\tilde{S}, \quad A_0 = \overline{\lambda}\tilde{F} + \lambda\tilde{S}, \quad A_1 = \lambda\tilde{F}, \tag{9}$$

where the matrices, $\tilde{S}$, $\tilde{F}$, $\tilde{N}$ (each of order $4M^2$) denote the change of phase together with a service success, service failure or no service attempt, respectively. For instance, the (i, j)-th entry of $\tilde{S}$ is the chance of a service success together with a change of phase from i to j (note that i and j are each 4-tuples). The sum $\tilde{S} + \tilde{F}$ is a stochastic matrix (as is evident from the construction below). Similarly, $\tilde{N}$ is a stochastic matrix. Hence the overall transition probability (infinite) matrix A is stochastic as well.

To construct $\tilde{S}$, $\tilde{F}$ and $\tilde{N}$, we define $M^2 \times M^2$ matrices $\tilde{S}_{k\ell}$, $\tilde{F}_{k\ell}$ and $\tilde{N}_{k\ell}$ for $k, \ell = 0, 1$. Taking $\tilde{S}_{k\ell}$ as an example, its (i, j)-th entry (each represented as a 2-tuple), describes the chance of a success together with a transition of belief state from i to j, when the environment of the first server is in state k and that of the second server is in state ℓ. Here i and j, each represent the overall system belief state in lexicographic order. That is, we should refer to i as (i_1, i_2) and similarly to j. A similar interpretation holds for $\tilde{F}_{\_k\ell}$ and $\tilde{N}_{\_k\ell}$. These aforementioned matrices are constructed (for $k, \ell = 0, 1$) as follows:

$$\begin{aligned}
\tilde{S}_{\_k\ell} &= \mu_\ell^{(2)} \Big(diag\big(vec(C')\big)\Big) (N^{(1)} \otimes S^{(2)}) + \mu_k^{(1)} \Big(diag\big(vec(\bar{C}')\big)\Big) (S^{(1)} \otimes N^{(2)}),\\
\tilde{F}_{\_k\ell} &= \bar{\mu}_\ell^{(2)} \Big(diag\big(vec(C')\big)\Big) (N^{(1)} \otimes F^{(2)}) + \bar{\mu}_k^{(1)} \Big(diag\big(vec(\bar{C}')\big)\Big) (F^{(1)} \otimes N^{(2)}),\\
\tilde{N}_{\_k\ell} &= (N^{(1)} \otimes N^{(2)}),
\end{aligned}$$

where $diag(\cdot)$ is an operation taking a vector and resulting in a diagonal matrix with the vector in the diagonal, $vec(\cdot)$ is an operation taking a matrix and resulting in a vector with the columns of the matrix stacked up one by one, and $\otimes$ is the standard Kronecker product.

To see the above, let us consider (for example) an element of the matrix $\tilde{S}_{\_k\ell}$ at coordinate $i = (i_1, i_2)$ and $j = (j_1, j_2)$. This describes the probability of the event

$$W = \{\text{Success of service together with a transition to belief state } (j_1, j_2)\},$$

where $X_1 = k$, $X_2 = \ell$, $\psi_1 = i_1$, and $\psi_2 = i_2$. The event W can be partitioned into W_1 (service attempt was on 1) and W_2 (service attempt was on 2). The chance of W_2 is C_{i_1,i_2}. With choosing Server 2 the success probability is $\mu_\ell^{(2)}$. Then under the event W_2, the belief state of Server 1 will be updated according to $N^{(1)}$ and the belief state of Server 2 with $S^{(2)}$. The $M^2 \times M^2$ matrix $diag(vec(C'))$ is a diagonal matrix where its diagonal elements are the rows of the matrix C, each represent the chance of $U = 2$.

With the matrices $\tilde{S}_{\_k\ell}$, $\tilde{F}_{\_k\ell}$ and $\tilde{N}_{\_k\ell}$ (for $k, \ell = 0, 1$) in hand, we construct the matrices $\tilde{S}$, $\tilde{F}$ and $\tilde{N}$ as:

$$\tilde{S} = (P^{(1)} \otimes P^{(2)}) \circledast \begin{bmatrix} \tilde{S}_{\_00} & 0 & 0 & 0 \\ 0 & \tilde{S}_{\_01} & 0 & 0 \\ 0 & 0 & \tilde{S}_{\_10} & 0 \\ 0 & 0 & 0 & \tilde{S}_{\_11} \end{bmatrix},$$

$$\tilde{F} = (P^{(1)} \otimes P^{(2)}) \circledast \begin{bmatrix} \tilde{F}_{\_00} & 0 & 0 & 0 \\ 0 & \tilde{F}_{\_01} & 0 & 0 \\ 0 & 0 & \tilde{F}_{\_10} & 0 \\ 0 & 0 & 0 & \tilde{F}_{\_11} \end{bmatrix},$$

$$\tilde{N} = (P^{(1)} \otimes P^{(2)}) \circledast \begin{bmatrix} \tilde{N}_{\_00} & 0 & 0 & 0 \\ 0 & \tilde{N}_{\_01} & 0 & 0 \\ 0 & 0 & \tilde{N}_{\_10} & 0 \\ 0 & 0 & 0 & \tilde{N}_{\_11} \end{bmatrix},$$

where $P^{(j)}$ for $j = 1, 2$ are the 2×2 probability transition matrices of the servers given by (2) and operation $\circledast$ is defined as

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \circledast \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} = \begin{bmatrix} a_{11}A & a_{12}A \\ a_{21}B & a_{22}B \end{bmatrix}.$$

Putting all of the above components together yields the probability transition matrix of $Z(t)$, A.

Stability Criterion A well-known sufficient condition for positive recurrence (stability) of QBDs such as $Z(t)$ is

$$\pi_\infty\big(A_1 - A_{-1}\big)\mathbf{1} < 0,$$

where π_∞ is the stationary distribution of the (finite) stochastic matrix $A_{-1} + A_0 + A_1$ and $\mathbf{1}$ is a column vector of ones. From (9), we see that this is also the stationary distribution of

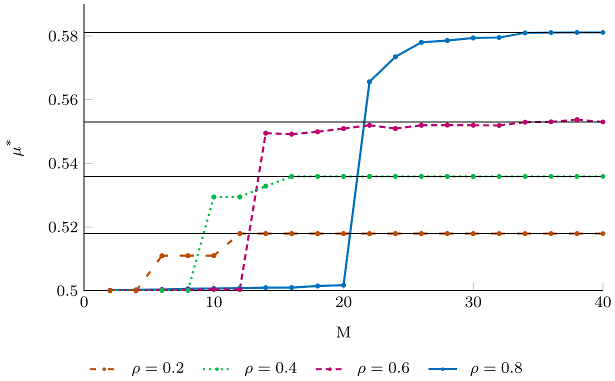

Fig. 7 Stability bound achieved by finite state controllers for observation scheme III with increasing M, computed by (10). The limiting horizontal lines are at μ^* computed by means of relative value iteration of Bellman equations

$\tilde{S} + \tilde{F}$ which does not depend on λ. This property of our QBD allows us to represent the stability criterion as

$$\lambda < \mu^* = \pi_\infty \tilde{S}\mathbf{1}, \tag{10}$$

with μ^* depending on the controller and system parameters but not depending on λ.

Numerical Illustration We now use our QBD model and the stability criterion (10) to explore the performance of finite state controllers.

In doing so, we consider the parameters as in (4) with $\rho_1 = \rho_2 = \rho$. Since in this situation, the servers are identical, the symmetric myopic policy is optimal and we thus restrict attention to a matrix C with

$$C_{i,j} = \begin{cases} 1, & i < j, \\ 0.5, & i = j, \\ 0, & i > j. \end{cases}$$

Using these parameters, we evaluated (10) for increasing M and for various values of ρ. The results are in Fig. 7. As expected, the performance of the finite state controller converges to that found by numerical solution of the Bellman equations as in the previous section. The sudden increase in performance as M increases (e.g. at $M = 20$ for $\rho = 0.8$) can be attributed to discretization phenomena. For reference, the values of μ^* obtained by Bellman equation (as well as the QBD when $M \to \infty$) are $0.5179,\ 0.5359,\ 0.5539$ and 0.5815 for $\rho = 0.2,\ 0.4,\ 0.6$ and 0.8, respectively.

6 Conclusion

This paper put forward methodology for relating information availability and system stability. Our focus was an elementary queueing model which by construction was designed to be as simple as possible. A key attribute is that the control decision influences the observation made. Explicit analysis of such systems is extremely challenging as is evident by both the complicated Bellman equations and the QBD structure that we put forward (even for a simple system as we consider). Nevertheless, insights obtained on the role of information are of interest.

Our model and numerical results, pave the way for explicit proofs of some of the structural properties outlined above. Moreover, the analysis remains to be extended to more general server environment models, as well as systems with more queues and control decisions. Related work is in Nazarathy et al. (2015), an earlier paper that leads to this work. An aspect in Nazarathy et al. (2015) that remains to be further considered is the networked case where the authors investigated (through simulation) cases in which the relationship of stability and throughput is not as immediate as in our current paper. A further related (recent) paper, Meshram et al. (2016), deals with a situation similar to our output observation case (III). In that paper, the authors consider the Whittle index applied to a similar system (without considering a queue and stability). Relating the Whittle index and system stability, is a further avenue that requires investigation.

Acknowledgements Azam Asanjarani's research is supported by the Australian Research Council Centre of Excellence for the Mathematical and Statistical Frontiers (ACEMS). Yoni Nazarathy's research is supported by ARC grant DP180101602.

References

Asanjarani A (2016) QBD Modelling Of a finite state controller for queueing systems with unobservable Markovian environments. In: Proceedings of the 11th International Conference on Queueing Theory and Network Applications, ACM, p 20

Baccelli F, Makowski AM (1986) Stability and bounds for single server queues in random environment. Stoch Model 2(2):281–291

Bäuerle N., Rieder U (2011) Markov Decision Processes with Applications to Finance. Springer Science & Business Media

Bramson M (2008) Stability of Queueing Networks. Springer

Cecchi F, Jacko P (2016) Nearly-optimal scheduling of users with Markovian time-varying transmission rates. Performance Evaluation 99(c):16–36. Elsevier Science Publishers BV

Chadès I, Chapron G, Cros MJ, Garcia F, Sabbadin R (2014) MDP Toolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. Ecography 37(9):916–920. Wiley Online Library

Halabian H, Lambadaris I, Lung CH (2014) Explicit characterization of stability region for stationary multi-queue multi-server systems. IEEE Trans Autom Control 59(2):355–370

Hernández-Lerma O, Lasserre JB (2012) Discrete-time Markov Control Processes: Basic Optimality Criteria. Springer

Johnston LA, Vikram K (2006) Opportunistic file transfer over a fading channel: a POMDP search theory formulation with optimal threshold policies. IEEE Trans Wirel Commun 5(2):394–405

Kella O, Whitt W (1992) A storage model with a two-state random environment. Operations Research 40(3):supplement-2:257–262

Koole G, Liu Z, Righter R (2001) Optimal transmission policies for noisy channels. Oper Res 49(6):892–899

Kuhn J, Nazarathy Y (2015) Wireless Channel Selection with Reward-Observing Restless Multi-armed Bandits. draft book chapter submitted

Larrañaga M, Ayesta U, Verloop IM (2014) Index policies for a multi-class queue with convex holding cost and abandonments. ACM SIGMETRICS Performance Evaluation Review, ACM 42(1):125–137

Larrañaga M., Assaad M, Destounis A, Paschos GS (2016) Asymptotically optimal pilot allocation over Markovian fading channels. arXiv:1608.08413

Latouche G, Ramaswami V (1999) Introduction to matrix analytic methods in stochastic modeling. SIAM

Leng B, Krishnamachari B, Guo X, Niu Z (2016) Optimal operation of a green server with bursty traffic, In: proceeding of Global Communications Conference (GLOBECOM), IEEE, pp 1–6

Li CP, Neely MJ (2013) Network utility maximization over partially observable Markovian channels. Perform Eval 70(7):528–548

Liu K, Zhao Q (2010) Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access. IEEE Trans Inf Theory 56(11):5547–5567

MacPhee IM, Jordan BP (1995) Optimal search for a moving target. Probab Eng Inform Sc 9(2):159–182

Meyn S (2008) Control techniques for complex networks. Cambridge University Press

Meshram R, Manjunath D, Gopalan A (2016) On the Whittle index for restless multiarmed hidden Markov bandits. arXiv:1603.04739

Mészáros A, Telek M (2014) Markov decision process and linear programming based control of MAP/MAP/n queues. In: European Workshop on Performance Engineering, 179–193 Springer

Nazarathy Y, Taimre T, Asanjarani A, Kuhn J, Patch B, Vuorinen A (2015) The challenge of stabilizing control for queueing systems with unobservable server states. In: 5th Australian Control Conference (AUCC), IEEE, pp 342–347

Puterman ML (2014) Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons

Sadeghi P, Kennedy RA, Rapajic BP, Shams R (2008) Finite-state Markov modelling of fading channels: A survey of principles and applications. IEEE Signal Process Mag 25(5):57–80

Smallwood RD, Sondik EJ (1973) The optimal control of partially observable Markov processes over a finite horizon. Oper Res 21(5):1071–1088

Tassiulas L, Ephremides A (1993) Dynamic server allocation to parallel queues with randomly varying connectivity. IEEE Trans Inf Theory 39(2):466–478

Verloop IM (2014) Asymptotically optimal priority policies for indexable and non-indexable restless bandits. Ann Appl Probab 26(4):1947–1995

Whittle P (1982) Optimization Over Time. John Wiley Sons Inc.

Whittle P (1988) Restless bandits: Activity allocation in a changing world. J Appl Probab 25:287–298

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.