Performance Evaluation and Policy Selection in Multiclass Networks*

Shane G. Henderson† Sean P. Meyn‡ Vladislav B. Tadić§

September 13, 2002

Abstract

This paper concerns modelling and policy synthesis for regulation of multiclass queueing networks.

A 2-parameter network model is introduced to allow independent modelling of variability and mean processing-rates, while maintaining simplicity of the model. Policy synthesis is based on consideration of more tractable workload models, and then translating a policy from this abstraction to the discrete network of interest.

Translation is made possible through the use of safety-stocks that maintain feasibility of workload trajectories. This is a well-known approach in the queueing theory literature, and may be viewed as a generic approach to avoid *deadlock* in a discrete-event dynamical system.

Simulation is used to evaluate a given policy, and to tune safety-stock levels. These simulations are accelerated through a variance reduction technique that incorporates stochastic approximation to tune the variance reduction. The search for appropriate safety-stock levels is coordinated through a cutting plane algorithm.

Both the policy synthesis and the simulation acceleration rely heavily on the development of approximations to the value function through fluid model considerations.

1 Introduction

This paper concerns policy synthesis and evaluation for large networks, as found in telecommunication and manufacturing applications. The proposed policies and our main conclusions are all based upon structural properties of Markov decision processes in this

*The authors would like to thank Xiren Cao for his careful reading of this paper.

†School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853, URL `http://www.orie.cornell.edu/∼shane` (shane@orie.cornell.edu). Work supported by NSF Grant DMI 0085165.

‡Coordinated Science Laboratory and the University of Illinois, 1308 W. Main Street, Urbana, IL 61801, URL `http://black.csl.uiuc.edu:80/∼meyn` (s-meyn@uiuc.edu). Work supported in part by NSF Grants ECS 99 72957 and DMI 00 85165

§Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia, URL `http://www.ee.mu.oz.au/people/Tadic.html` (v.tadic@ee.mu.oz.au).

special setting, and various approximation techniques to reduce complexity of the models and the associated control solutions.

The issues addressed here have many analogs in the Discrete Event Dynamical System (DEDS) literature. The now standard formulation of a DEDS is a discrete-state process, whose transitions arise due to uncontrolled events. Desirable performance is obtained by choosing parameters in the system appropriately, or imposing a *supervisor* to disable undesirable transitions. Important goals include deadlock avoidance, and ensuring that the system can reach desirable target states from any initial configuration.

The goals are similar in resource allocation for effective network management. One may take as a state-space the set of buffer levels, which is typically discrete. Events are triggered by arrivals of customers, arrival of demand, or the completion of processing at some station in the network. A *policy* determines which events may take place in a time interval by specifying which of several activities are in progress at a given time-instance. A situation akin to deadlock can arise if sufficient material is not available to perform any activity at some group of stations in the network.

It would appear that resource allocation in a queueing network could be addressed by studying these analogies, and appealing to results from the DEDS literature. One can also view the network as a Markov decision process, and use the tools from that arena to identify an optimal policy. However, such approaches suffer greatly from the "curse of dimensionality" in dealing with complex models. Furthermore, it is often the case that optimal policies are extremely complex, and therefore difficult to implement. Finally, in a queueing model there is additional structure that enables policy synthesis through topological consideration, and dynamic programming arguments for various relaxations of the primary network model. For these reasons, in this paper we avoid combinatorial or linguistic approaches to policy synthesis.

Our goal is to identify policies that are not necessarily optimal, but that have other desirable properties. Characteristics of an effective policy include moderate complexity; flexibility in the face of non-standard constraints; and the possibility of adaptation to a dynamic environment. In this paper we develop one approach to policy synthesis based on first identifying a class of policies from consideration of a deterministic fluid model, and then using simulation to identify an efficient policy for the original discrete model from within this class. This approach is similar in spirit to that adopted in much recent work on Brownian network approximations, where again one constructs a policy for a *stochastic* fluid model and then translates this to the original discrete network. Under certain conditions on the model, it is known that a policy obtained in this fashion is "asymptotically optimal" in heavy traffic [23, 4, 49]. In some cases the optimal policy for a fluid approximation is relatively easily identified [22, 49]. Specialized numerical approaches may prove helpful in situations where this is not the case (see e.g. [36, 18]).

Both the fluid model and the Brownian network approximations mentioned above rely on simplifying the incredibly complex dynamics of a fully general network model to a more manageable level. We focus on the *workload* processes of certain resources in the network. When these resources are heavily loaded, the workload levels evolve at a much slower speed than the process giving the occupancy of the buffers in the network. This "two time scale" behavior leads to some key simplifications, and is very similar to that encountered in nearly-decomposable systems [14]. We are then able to define a policy for

the (fluid) workload model.

There are several issues involved in translating a fluid policy back to a discrete network.

The fluid policy often includes "switching curves" where the policy takes on different forms on different sides of these curves. The optimal switching curves for a stochastic model may be far more 'conservative' than those obtained through the optimization of a fluid network model [12], and so one may wish to "tune" the switching curves. We do not consider this issue further here, since sensitivity of cost with respect to an interior switching curve is typically very low, especially when switching surfaces are not 'attracting' (see [12]).

Performance is typically far more sensitive to *safety stocks*, which can be described as follows. When translating a policy from a deterministic fluid setting to a stochastic setting, one needs to take care with variability. Suppose, for example, that the policy requires that a station serve jobs of a particular class for a certain period of time. In a deterministic setting, one can guarantee that the station will not run out of work, but this is no longer true in a stochastic setting. Safety stocks are defined as target levels of inventory at a buffer that the system attempts to maintain to avoid idle-time at a station.

Consider, for the sake of illustration, the network in Figure 1. When the processing rates at the exit buffers are relatively slow it is known that the optimal policy for the fluid model gives priority to exit buffers, while avoiding starvation at each resource (station). An optimal policy for a particular set of parameters is given in [48] for an associated stochastic-discrete model. It is found that the form of the optimal policy is similar: At Station 1, priority is given to buffer 4 over buffer 1 unless the content of buffer 2 is lower than a certain safety-stock level, in which case priority goes to buffer 1. The policy at Station 2 is entirely analogous. Although the optimal policy takes this form, the sensitivity of performance to safety-stock levels is very high. If the specified level of safety-stock at buffer 2 is too low, then buffer 2 will be starved for work, leading to very high levels of congestion, or even instability. An example of instability with 2 parallel servers is given in [23].

In heavy traffic one can argue that precise values of safety-stock levels are unimportant. In an academic example, values of order $\log((1-\rho)^{-1})$ are found to be asymptotically optimal in heavy-traffic [4], and these results are extended to general network models in [49]. It is known that the steady-state customer population is of order $(1-\rho)^{-1}$, so the relative cost incurred from the safety-stock values become negligible in heavy traffic, where $\rho \sim 1$.

In practice, the choice of safety-stock levels can be critical, especially when variability is high. If safety-stock levels are too low then instability can result. If safety-stock levels are too high then the network incurs unnecessarily high costs.

In this paper we focus on the issue of safety-stock selection. We propose the use of simulation to explore the relative performance of a family of policies indexed by safety-stock levels. The complexity of the full discrete network model makes this search problematical. Furthermore, the networks we consider are often heavily loaded, and this is precisely the setting in which very long simulation runlengths are required to achieve reasonable accuracy [1, 61]. Variance reduction techniques such as those developed in [26, 27] can reduce the required simulation runlengths substantially. Unfortunately, there are some difficulties in applying these techniques to non-Markovian network models [24].

For these reasons we introduce a new class network models. The *2-parameter network models* described in Section 2.1 may be motivated through an analysis of "worst-case" large deviations behavior. The primary motivation is similar in spirit to the Brownian models: In both the 2-parameter and Brownian models one may prescribe drift and volatility of network processes, while maintaining a compact set of evolution equations, and in both cases the queue-length process may be viewed as a controlled Markov process. They possess an advantage over Brownian models in that performance evaluation is straightforward using simulation, and there are no issues related to existence or nonexistence of the underlying stochastic processes. Moreover, for a wide range of operating policies, the Markov property allows us to extend the variance reduction techniques of [26, 27] to the class of 2-parameter models.

The variance reduction technique investigated in this paper is basically a control variate method. In this technique, one adds a zero-mean term that is negatively correlated with one's estimator. Since the newly introduced term has zero mean, the controlled estimator has the same mean as the original estimator, and the negative correlation serves to "dampen" variability in the estimator that would otherwise lead to a large variance. Our control variates rely on approximation of the value function, which is also key to policy selection. See [39] for more details on control variates.

It turns out that under very general conditions there exists a family of control variates indexed by a (possibly multivariate) parameter r. Each value of the parameter yields a control variate with zero mean. One might then wish to choose the parameter r^* that yields the greatest variance reduction. We develop a stochastic gradient algorithm for estimating r^* adaptively within the simulation, and demonstrate its effectiveness in a 2-parameter model of the single-server queue.

Our proposed methodology for identifying appropriate safety-stocks involves an optimization. To reduce the work required from an exhaustive search we apply a cutting plane algorithm. The success of a cutting plane algorithm relies on convexity of the objective function and the ability to estimate the function's gradient. Such convexity was empirically demonstrated through a variety of examples in [28] and [12]. Subgradients are needed to successfully apply cutting plane methods. To obtain these, one would like to employ a method such as perturbation analysis but, as shown in [28], gradient estimators based on perturbation analysis will require prohibitive computation. We instead use finite differences.

The remainder of the paper is organized as follows. Section 2 contains a review of modelling techniques, and introduces the 2-parameter model. Recent results concerning optimization are surveyed in Section 2.3 following [48, 49, 12]. This section also describes methods for translating a policy from one model to another using safety-stocks.

Section 3 surveys the use of certain variance reduction techniques and shows how to adaptively tune the variance reduction using a 2 timescale algorithm. Section 4 discusses the application of cutting plane methods to identify appropriate safety-stock levels, and conclusions and recommendations for future research are given in Section 5.

2 Network models & optimal control

Management of a production system or a communications network may be cast as a regulation problem for an associated stochastic model. However, to solve this regulation problem one must balance realism and tractability of the stochastic model. The choice of an appropriate model for policy selection will be very different than the best model for a simulation. In this section we describe some basic models and the relationships between them, and review some recent structural results concerning optimal control of network models.

2.1 The 2-parameter stochastic model

Our notion of a multiclass network consists of ℓ_b buffers, ℓ_a activities, and ℓ_r resources. Buffer levels are assumed to be nonnegative, and we may also impose upper bounds. We assume throughout that $\ell_a \geq \ell_b$.

We let $\boldsymbol{Q} = \{Q(t) : t \geq 0\}$ denote the ℓ_b vector of buffer levels as a function of time. A general model commonly considered in the literature is described by the following continuous-time equations,

$$Q_i(t) = x_i - S_i(Z_i(t)) + A_i(t) + \sum_{j=1}^{\ell_a} R_{ij}(Z_j(t)) \,, \tag{1}$$

where $Q(0) = x$ denotes the initial configuration of buffer levels. The *allocation process* $\boldsymbol{Z}_j = (Z_j(t) : t \geq 0)$ gives the cumulative time allocated to activity j by time t, so that $Z_j(0) = 0$, $j = 1, \ldots, \ell_a$.

The nondecreasing, integer-valued function $\boldsymbol{S}_i$ represents random service times. For all $1 \leq i \leq \ell_b$ and all $1 \leq j \leq \ell_a$, the function $\boldsymbol{R}_{ij}(\,\cdot\,)$ represents internal transfers to buffer i due to activity j. effects of a combination of possibly uncontrolled, possibly random routing, and random service times. The process $\boldsymbol{A}_i = \{A_i(t) : t \geq 0\}$ denotes exogenous arrivals to buffer i for $1 \leq i \leq \ell_b$. It is typically assumed that all of the driving processes $\{\boldsymbol{R}_{ij}, \boldsymbol{S}_j, \boldsymbol{A}_i\}$ are defined via independent renewal processes.

Because of this generality, the model (1) typically fails to provide a tractable model for policy selection or simulation. This explains only partially our motivation for seeking a simpler model of the dynamics of $\boldsymbol{Q}$:

(i) A considerable amount of detailed information is required in the construction of the model (1). In particular, one must completely specify processing time distributions and interarrival time distributions. Such information may not be readily available in a given application.

(ii) It will often be the case that the networks under consideration are moderately to heavily loaded. In such settings, a Brownian model can be rigorously motivated through heavy traffic arguments; see, for example, [55, 10, 62, 9]. Brownian models depend primarily on the first and second moments of the building-block processes $\{\boldsymbol{R}, \boldsymbol{S}, \boldsymbol{A}\}$, and not on finer distributional information.

(iii) The simulation runlengths required to achieve a specified accuracy in estimates of mean queue length and other moments increase rapidly with increasing load on the resources in many stochastic models [1, 61, 27]. This issue becomes especially acute in the context of this paper since many competing policies must be simulated, and each will require large simulation runlengths if the network load is significant. Variance reduction techniques can be used to reduce this computational burden for certain classes of models [26, 27]. However, so far the extensions of these methods to the case of general renewal-type statistics has proved cumbersome [24].

So, we are motivated to consider alternative models for network dynamics. There is a need for a network model of low complexity that can capture the essential features of the physical network based on general service and arrival statistics, and that is amenable to efficient performance evaluation techniques through simulation

The Brownian models advanced in, for example, [37, 11, 9, 23] are an elegant class of models that are able to capture both the mean and variability in the building-block processes of the network. However, a Brownian model may also suffer from significant complexity. Furthermore, there are some discrete network models for which the "natural" Brownian network model does not exist [15].

We propose here a class of models that are similar in concept to the Brownian network models in that they attempt to capture both mean and variability in primitive network processes. We begin with a discussion of the M/M/1 queue.

The stochastic process giving the number of customers in an M/M/1 queue (the buffer size) with arrival rate α and service rate μ can be represented as a *discrete-time* process after uniformizing the underlying continuous time Markov chain. Let $\boldsymbol{Q}$ denote the one-dimensional buffer process; $\boldsymbol{S}$ an i.i.d. sequence of random variables that reflect potential customer service completions; and let $\boldsymbol{A}$ denote an i.i.d. sequence representing exogenous arrivals. For $k \geq 0$ we write,

$$Q(k+1) = Q(k) - S(k+1)U(k) + A(k+1), \tag{2}$$

where $\boldsymbol{U}$ is a one-dimensional *allocation sequence*. We assume that it is defined by the *non-idling policy*, $U(k) = \mathbb{I}(Q(k) > 0)$, $k \geq 1$. The stochastic process $\boldsymbol{A}$ is Bernoulli with $\mathsf{P}\{A(k) = 1\} = \alpha(\alpha+\mu)^{-1}$, and since an event occurs at each time point in the uniformized process, $S(k) = 1 - A(k)$, $k \geq 1$.

More general networks in which all primitive processes follow independent Poisson processes may be uniformized, leading to an expression similar to (2) for the network dynamics. This leads to a countable state space, controlled Markov chain model with state process $\boldsymbol{Q}$. This class of network models is essentially a *1-parameter family*, in that the dynamics of the process are completely determined by the *rates* (α, μ above) of the interarrival and service time processes. Basically, each primitive process (arrival, service) is required to follow a Poisson process that is determined by its rate, which is a single parameter. Therefore, it is possible to model a given drift, but one must then accept the variability of a Poisson process with the given drift.

To capture variability one can use Erlang distributions, but it is then necessary to expand the state space to retain the Markov property. We prefer a different approach.

By relaxing the requirement that the elements of $Q(k)$ be integer-valued we may retain an ℓ_b-dimensional Markov model, and prescribe mean and variance independently. This non-integrality is the primary difference between the 1-parameter model as given by a uniformized Markov chain and the 2-parameter models introduced next.

The 2-Parameter Model
For given first and second order statistics of the network, this is defined by the recursion,

$$Q(k+1) = Q(k) + B(k+1)U(k) + A(k+1), \qquad Q(0) = x\,. \tag{3}$$

The following assumptions are imposed on the policy and parameters:

(i) $\boldsymbol{B}$ is an i.i.d. sequence of $\ell_b \times \ell_a$ matrices; and $\boldsymbol{A}$ is an i.i.d. sequence of ℓ_b-dimensional vectors. The mean and variance of $\boldsymbol{A}$ and $\boldsymbol{B}$ are user-specified parameters.

(ii) The allocation sequence $\boldsymbol{U}$ is adapted to $(\boldsymbol{Q}, \boldsymbol{A}, \boldsymbol{B})$, and satisfies the linear constraints,

$$U(k) \ge \boldsymbol{\theta}\,, \quad CU(k) \le \boldsymbol{1}\,, \qquad k \ge 1,$$

where C is the $\ell_r \times \ell_a$ *constituency matrix*; $\boldsymbol{1}$ is the ℓ_r-dimensional vector of ones; and $\boldsymbol{\theta}$ is the ℓ_a-dimensional vector of zeros.

(iii) The queue length process $\boldsymbol{Q}$ is similarly constrained, $Q(k) \in \mathsf{X}$, $k \ge 0$, where $\mathsf{X} \subset \mathbb{R}_+^{\ell_b}$ is a polyhedron representing both positivity constraints, and bounds on buffer levels if present.

These models are "2-parameter" in the sense that the user specifies both the mean and variance of the matrices $B(k)$ and vectors $A(k)$, $k \ge 1$. By way of contrast, the M/M/1 queue as specified earlier was 1-parameter in the sense that the distributions were restricted to be Bernoulli, so that each distribution was specified by a single parameter (the probability of "success"). This restriction arises because the M/M/1 queue is built from two Poisson processes, each of which is specified by a single parameter. But 2-parameter models can also capture (in a certain sense) more general distributions than exponential. For instance, as noted above it is often assumed that the sequence of processing times of jobs at a particular buffer follows a renewal process. Let us consider how a renewal process can be approximated within the 2-parameter framework.

Suppose that $\widetilde{\boldsymbol{N}} = (\widetilde{N}(t) : t \ge 0)$ is a delayed renewal process with interevent times $Y_0, Y_1, Y_2, \ldots$, so that $Y_0, Y_1, \ldots$ are independent and $Y_1, Y_2, \ldots$ are i.i.d. with mean m and variance $\sigma^2 < \infty$. Then, the law of large numbers and central limit theorem hold, i.e.,

$$\tfrac{1}{t}\widetilde{N}(t) \to m^{-1}, \text{ a.s., and } t^{1/2}\Big(\tfrac{1}{t}\widetilde{N}(t) - \tfrac{1}{m}\Big) \Rightarrow N(0, \sigma^2/m^3)\,,$$

as $t \to \infty$, where " $\Rightarrow$ " denotes weak convergence and $N(p_1, p_2)$ a normal random variable with mean p_1 and variance p_2 (see [57] for example). We wish to construct a 2-parameter process $(N(k) : k \ge 0)$ that shares these asymptotic properties.

Let $N(0) = 0$, and for $k \geq 1$ let $N(k) = \sum_{i=1}^{k} B(i)$, where $\boldsymbol{B} = (B(i) : i \geq 1)$ consists of i.i.d. nonnegative random variables. Given a sampling increment $T_s > 0$ we wish to choose the distribution of $\boldsymbol{B}$ so that the random variables $\widetilde{N}(kT_s)$ and $N(k)$ have approximately the same distribution for $k \geq 1$. Consideration of large k suggests the two restrictions,

$$\mathsf{E}[B(1)] = \frac{T_s}{m} \qquad \textit{and} \quad \mathrm{Var}\,[B(1)] = \frac{T_s \sigma^2}{m^3}.$$

Note that $\boldsymbol{N}$ is not an exact representation of $\widetilde{\boldsymbol{N}}$, but rather, it shares the asymptotic properties of $\widetilde{\boldsymbol{N}}$.

This building block can be used to build a 2-parameter model that approximates a given general model (1) if the primitive processes $\boldsymbol{R}, \boldsymbol{S}, \boldsymbol{A}$ are defined via renewal processes.

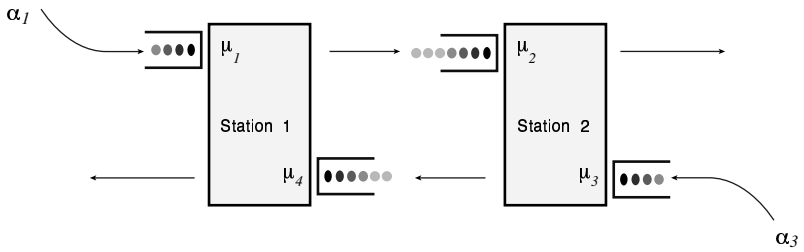

Figure 1: The Kumar-Seidman-Rybko-Stolyar (KSRS) network.

To help unify the discussion and clarify the methodology we use the Kumar-Seidman-Rybko-Stolyar (KSRS) network depicted in Figure 1 as a running example. Suppose that the arrival processes of jobs, and the service processes at each of the 4-buffers are independent renewal processes. We can model this system as a 2-parameter model by taking

$$B(k) = \begin{bmatrix} -S_1(k) & 0 & 0 & 0 \\ S_1(k) & -S_2(k) & 0 & 0 \\ 0 & 0 & -S_3(k) & 0 \\ 0 & 0 & S_3(k) & -S_4(k) \end{bmatrix}, \quad A(k) = \begin{bmatrix} A_1(k) \\ 0 \\ A_3(k) \\ 0 \end{bmatrix}.$$

All of these variables are i.i.d. and mutually independent, with $\mathsf{E}[S_i(k)] = \mu_i$ and $\mathsf{E}[A_i(k)] = \alpha_i$, $k \geq 1$. It is assumed that $U_1(k) + U_4(k) \leq 1$ and $U_2(k) + U_3(k) \leq 1$. These activity constraints at each station are summarized by the selection of the constituency matrix,

$$C = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

The model (3) is well-motivated through our goal to maintain low complexity. Furthermore, under appropriate assumptions, in heavy-traffic it approximates the original stochastic model since both are approximated by a Brownian model. Indeed, in heavy traffic, the precise form of the building-block distributions is typically unimportant; see Section 3.1. But how should we choose the random matrix $\boldsymbol{B}$ and random vector $\boldsymbol{A}$ subject to the first and second order statistical constraints?

Consider first the construction of a random variable with given mean μ. One possibility is the discrete random variable,

$$D = \begin{cases} \overline{\mu} & \text{with probability } p; \\ 0 & \text{with probability } 1-p, \end{cases} \tag{4}$$

where $\overline{\mu} \geq \mu$, and $p = \mu/\overline{\mu}$. If $\overline{\mu}$ is a given deterministic upper bound on the random variable D, then this choice gives the random variable with the highest variance, and the worst-case large deviations asymptotics (see [16, Lemma 2.4.1]).

In some of the numerical experiments below we consider distributions of this form, parameterized by a variability parameter $\kappa > 0$. Letting η denote the coefficient of variation for D when $\kappa = 1$ we take

$$\overline{\mu} = (1 + \eta^2\kappa)\mu\,, \qquad p = \mu/\overline{\mu}.$$

This gives the required mean, and for all $\kappa > 0$ the squared coefficient of variation is given by,

$$\frac{\mathrm{Var}\,[D]}{\mathsf{E}[D]^2} = \kappa\eta^2.$$

When $\kappa = 0$ this is a discrete-time version of the *fluid model* in which there is no variability. This special case is central to control design and will be investigated next.

2.2 Workload models

One can develop a (continuous time) fluid analog of (1) that is the starting point for model reduction for policy synthesis. We begin with the simplest instance in which there is no variability.

The (continuous-time) fluid model is associated with a given 2-parameter model. To motivate its form, observe that (3) can be written as

$$Q(k) = Q(0) + \sum_{j=1}^{k} B(j)U(j-1) + \sum_{j=1}^{k} A(j), \qquad k \geq 1.$$

The Fluid Model
Associated with a given 2-parameter model, this is described by the continuous-time equation,

$$q(t) = x + Bz(t) + \alpha t\,, \qquad t \geq 0. \tag{5}$$

The following assumptions are imposed on the policy and parameters:

(i) The $\ell_b \times \ell_a$ matrix B, and the ℓ_b-dimensional vector α are the means of $B(k)$, $A(k)$ respectively.

(ii) The cumulative allocation $\boldsymbol{z}$ satisfies the linear constraints,

$$\zeta(t) = \frac{d}{dt} z(t) \in \mathsf{U} := \{u \in \mathbb{R}_+^{\ell_a} : Cu \leq \boldsymbol{1}\}\,, \qquad t \geq 0.$$

(iii) The queue length process $\boldsymbol{q}$ is similarly constrained, $q(t) \in \mathsf{X}$, $t \geq 0$, where X is a polyhedron $\mathsf{X} \subset \mathbb{R}_+^{\ell_b}$.

Policy synthesis for the fluid model and methods of translation to discrete models are developed in [54, 11, 41, 42, 3, 43, 48, 60]. Conditions guaranteeing asymptotic optimality of such policies are given in [49, 12]. Although we do not require the full machinery developed in these papers to identify effective policies for the examples presented in this paper, for complex models this machinery will prove essential. We give a brief sketch of the key ideas in Section 2.3.

The fluid model may be regarded as a differential inclusion, $\frac{d}{dt}q \in \mathsf{V}$, where '$\frac{d}{dt}$' denotes the right derivative, and

$$\mathsf{V} = \{Bu + \alpha : u \in \mathsf{U}\}\,.$$

The velocity set V is a polyhedron since U is a polyhedron. In [49] it is constructed as the general form $\mathsf{V} = \{v \in \mathbb{R}^{\ell_b} : \langle \xi^i, v\rangle \geq -(\delta_i - \rho_i),\ i = 1, \ldots, \ell_v\}$ for some integer ℓ_v, where $\langle \xi^i, v\rangle$ denotes the dot product of the vectors ξ^i and v. Each of the constants $\{\delta_i\}$ is equal to 0 or 1. If $\delta_i = 1$ then we call $\xi^i \in \mathbb{R}^{\ell_b}$ a *workload vector*. Intuitively speaking, ξ^i is a workload vector if there is a resource corresponding to ξ^i that can be a bottleneck in certain situations; see [49] for details. Define $\rho_i = \langle \xi^i, \alpha\rangle$, and the *system load* by $\rho = \max \rho_i$.

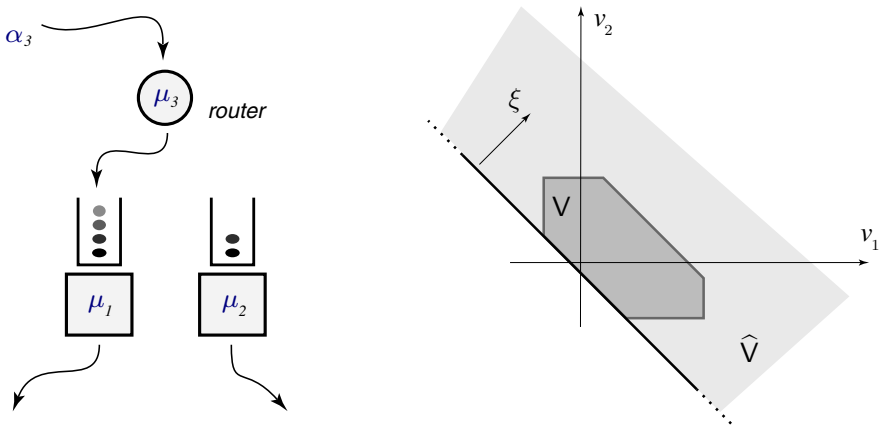

Figure 2: At left is the simple routing model, and at right its velocity set V restricted to velocity vectors satisfying $v_3 = 0$.

Consider for example the routing model shown in Figure 2, whose fluid model is given by (5) with,

$$B = \begin{bmatrix} -\mu_1 & 0 & \mu_3 & 0 \\ 0 & -\mu_2 & 0 & \mu_3 \\ 0 & 0 & -\mu_3 & -\mu_3 \end{bmatrix}, \quad \alpha = \begin{bmatrix} 0 \\ 0 \\ \alpha_3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

We have four workload vectors, and corresponding loads $\{\rho_i = \langle \xi^i, \alpha \rangle\}$ given by

$$\begin{array}{rclcrcl} \xi^1 &=& (\mu_1+\mu_2)^{-1}(1,1,1)^{\scriptscriptstyle T}, & & \rho_1 &=& \alpha_3(\mu_1+\mu_2)^{-1} \\ \xi^2 &=& (m_1,0,0)^{\scriptscriptstyle T}, & & \rho_2 &=& 0 \\ \xi^3 &=& (0,m_2,0)^{\scriptscriptstyle T}, & & \rho_3 &=& 0 \\ \xi^4 &=& (0,0,m_3)^{\scriptscriptstyle T}, & & \rho_4 &=& \alpha_3/\mu_3\,, \end{array}$$

where $m_i = 1/\mu_i$. We see that ξ^1 defines the workload at the two downstream stations, pooled together to form a single resource.

The respective loads are given by

$$\rho_1 = \alpha_3/(\mu_1+\mu_2),\ \rho_2 = \rho_3 = 0,\ \rho_4 = \alpha_3/\mu_3.$$

The system load is $\rho = \max(\rho_1, \rho_4) = \rho_1$ under the assumption that the router is relatively fast, so that $\mu_3 > \mu_1 + \mu_2$.

The fluid model for the KSRS model is given by (5) with

$$B := \mathsf{E}[B(k)] = \begin{bmatrix} -\mu_1 & 0 & 0 & 0 \\ \mu_1 & -\mu_2 & 0 & 0 \\ 0 & 0 & -\mu_3 & 0 \\ 0 & 0 & \mu_3 & -\mu_4 \end{bmatrix}, \quad \alpha := \mathsf{E}[A(k)] = \begin{bmatrix} \alpha_1 \\ 0 \\ \alpha_3 \\ 0 \end{bmatrix},$$

In scheduling models of this form with two stations, the two workload vectors and associated load parameters are given explicitly by

$$\boldsymbol{\rho} = -CB^{-1}\alpha\,, \qquad [\xi^1, \xi^2]^{\scriptscriptstyle T} = -CB^{-1}\,.$$

Consequently, in this example the workload vectors and load parameters are given by,

$$\xi^1 = \begin{bmatrix} m_1 \\ 0 \\ m_4 \\ m_4 \end{bmatrix}, \quad \xi^2 = \begin{bmatrix} m_2 \\ m_2 \\ m_3 \\ 0 \end{bmatrix}, \qquad \begin{array}{rcl} \rho_1 &=& m_1\alpha_1 + m_4\alpha_3 \\ \rho_2 &=& m_2\alpha_1 + m_3\alpha_3 \end{array}$$

with $m_i = \mu_i^{-1}$ for $i = 1, 2, 3, 4$.

The *workload process* for a fluid model is the vector process $(w(t) : t \geq 0)$ where $w(t) = \Xi q(t)$, and $\Xi = [\xi^1 | \cdots | \xi^{\ell_v}]^T$ is a matrix whose rows consist of workload vectors. Thus, the workload process is a linear transformation of the buffer levels that tracks the workloads of resources in the network.

It is often the case that the *important* dynamics of the model are captured by a subset of the components of the workload process. We usually focus on the most heavily loaded resources. Accordingly, suppose that the vectors $\xi^1, \ldots, \xi^{\ell_v}$ are ordered so that $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_{\ell_v}$. The nth workload relaxation ($n \leq \ell_v$) is the n-dimensional workload process $(\widehat{w}(t) : t \geq 0)$ where $\widehat{w}(t) = \widehat{\Xi}\widehat{q}(t)$, and $\widehat{\Xi} = [\xi^1 | \cdots | \xi^n]^T$. Provided the vectors $\{\xi^i : 1 \leq i \leq n\}$ are linearly independent, the workload process is subject to the *decoupled* constraints, $\frac{d}{dt}\widehat{w}_i \geq -(1-\rho_i)$, $i = 1, \ldots, n$.

The nth workload relaxation can also be viewed in "buffer coordinates". This is described by the differential inclusion $\frac{d}{dt}\widehat{q} \in \widehat{\mathsf{V}}$, where the relaxed velocity set is given by $\widehat{\mathsf{V}} := \{v \in \mathbb{R}^\ell : \langle \xi^i, v\rangle \geq -(1-\rho_i), \quad 1 \leq i \leq n\}$. This is a true relaxation in the sense that $\mathsf{V} \subset \widehat{\mathsf{V}}$.

Workload relaxations are analogous to *state space collapse* seen in analysis of network models in heavy traffic, where the load is close to unity [56, 21, 8, 62]. The idea is that only some of the constraints in V are important from the point of view of policy synthesis, and removing less restrictive constraints leads to a far simpler control problem.

Consideration of the geometry of the velocity set shown in Figure 2 suggests that a one-dimensional relaxation is suitable for the routing model, with $\widehat{\mathsf{V}} = \{v \in \mathbb{R}^3 : \xi^1 \cdot v \geq -(1-\rho_1)\}\}$. In the KSRS example a two-dimensional relaxation is appropriate if the loads at the two stations are comparable.

A *probabilistic workload model* is obtained by introducing additive noise to the nth workload relaxation $(\widehat{w}(t) : t \geq 0)$,

$$\widehat{W}(t) = \widehat{W}(0) - (\boldsymbol{1} - \boldsymbol{\rho})t + I(t) + N(t). \tag{6}$$

Here $\boldsymbol{N}$ is an n-dimensional stochastic process with zero mean and finite variance. In this equation $\boldsymbol{I}$ is an n-dimensional idleness process that ensures that $\widehat{W}(t) \geq 0$ for all t. It is determined by the particular policy. Its components are non-decreasing, and adapted to $\boldsymbol{N}$ and $\widehat{\boldsymbol{W}}$. Note that the deterministic workload model defined earlier is precisely (6) in the special case where $\boldsymbol{N}$ is set to zero.

The Brownian model is obtained on choosing $\boldsymbol{N}$ as a Brownian motion on $\mathbb{R}^n$. While this model is valuable for obtaining structural insights for optimal policies, to construct a simple probabilistic workload model for simulation it is convenient to return to the discrete time setting.

Consider again the KSRS model for the sake of illustration. Letting $W(k) = \widehat{\Xi}Q(k)$ where $\widehat{\Xi} = [\xi^1 | \xi^2]^T$ and $(Q(k) : k \geq 0)$ is the 2-parameter model we obtain,

$$\begin{pmatrix} W_1(k+1) \\ W_2(k+1) \end{pmatrix} = \begin{pmatrix} W_1(k) \\ W_2(k) \end{pmatrix} \quad - \begin{pmatrix} m_1 S_1(k+1) U_1(k) + m_4 S_4(k+1) U_4(k) \\ m_2 S_2(k+1) U_2(k) + m_3 S_3(k+1) U_3(k) \end{pmatrix} + \begin{pmatrix} m_1 A_1(k+1) + m_4 A_3(k+1) \\ m_2 A_1(k+1) + m_3 A_3(k+1) \end{pmatrix}.$$

Assume that the coefficient of variation η_i of the service time S_i is independent at a given station, so that $\eta_1 = \eta_4$ and $\eta_2 = \eta_3$. In this case, a relaxation is obtained on observing that $m_i S_i(k)$ and $m_j S_j(k)$ have identical second-order statistics for $k \geq 1$, with common mean equal to -1, with $(i, j) = (1, 4)$ or $(2, 3)$.

This motivates a workload relaxation of the form,

$$\widehat{W}(k+1) = \widehat{W}(k) - D(k+1) + I(k), \qquad k \geq 0\,,$$

where $\boldsymbol{I}$ is again the idleness process, and $\boldsymbol{D}$ is i.i.d.. To maintain second order statistics we may take,

$$D(k) := \begin{pmatrix} m_1 S_1(k) \\ m_2 S_2(k) \end{pmatrix} - \begin{pmatrix} m_1 A_1(k) + m_4 A_3(k) \\ m_2 A_1(k) + m_3 A_3(k) \end{pmatrix}, \qquad k \geq 1.$$

This discrete-time process is i.i.d. with common mean and covariance given by

$$\begin{aligned} \mathsf{E}[D(k)] &= (1-\rho_1, 1-\rho_2)^{\scriptscriptstyle T}; \\ \mathrm{Var}\,[D(k)] &= \begin{bmatrix} \eta_1^2 + \sigma_{A_1}^2 m_1^2 + \sigma_{A_3}^2 m_4^2\,, & \sigma_{A_1}^2 m_1 m_2 + \sigma_{A_3}^2 m_3 m_4 \\ \sigma_{A_1}^2 m_1 m_2 + \sigma_{A_3}^2 m_3 m_4\,, & \eta_2^2 + \sigma_{A_1}^2 m_2^2 + \sigma_{A_3}^2 m_3^2 \end{bmatrix} \end{aligned} \tag{7}$$

Alternatively, we may choose a simpler discrete random variable with the same mean and covariance. One possibility is the general form,

$$D(k) = \begin{cases} \bar{d}^1 & \text{with probability } p; \\ \bar{d}^2 & \text{with probability } q; \\ 0 & \text{with probability } 1-p-q. \end{cases}$$

The probabilities $\{p, q\}$ and the two-dimensional vectors $\{\bar{d}^i\}$ can be chosen to satisfy the required second order statistics (7).

2.3 Structure of optimal policies

We now turn to application of these models to the policy synthesis problem. For concreteness, suppose that our goal is to minimize inventory, given a prescribed set of demand and arrival rates. In the academic literature on network management this is typically posed as a dynamic optimization problem, usually for a stochastic model such as (1). A popular choice is the average-cost optimization criterion,

$$\textbf{min} \lim_{t\to\infty} \mathsf{E}[c(Q(t))],$$

with c the ℓ_1-norm, and where the minimum is over all policies. While we cannot expect to compute an exact optimizer, we can obtain an effective policy with desirable properties through insight obtained from the analysis of this optimization problem. Moreover, the procedure described here leads to an approximately optimal policy under general assumptions on the model.

In the remainder of this section we survey structural properties of optimal polices for the 2-parameter model, and some algorithms for constructing effective polices based on these results.

The solution to the average cost optimization problem is of *state-feedback* form for the 2-parameter model. We have $U(k) = g_*(Q(k))$, $k \ge 0$, where the feedback law g_* together with the optimal cost γ_*, and a relative value function h_* solve the following dynamic programming equations:

$$\begin{aligned} h_*(x) &= \textbf{min}\ \mathsf{E}\Big[\sum_{k=0}^{\tau_\theta - 1} [c(Q(k)) - \gamma_*] \mid Q(0) = x\Big] \\ g_*(x) &= \arg\min_{u\in\mathsf{U}} \mathsf{E}[h_*(x + B(1)u + A(1))] \end{aligned} \tag{8}$$

In the first equation, the minimum is over all allocation sequences $\boldsymbol{U}$, and the hitting time $\tau_\theta = \min\{k > 0 : Q(k) = \boldsymbol{\theta}\}$ is the first time $k > 0$ that the network is *empty*. Under

mild conditions on the network these equations admit a solution, and the resulting control sequence using g_* is average-cost optimal (see [46, 19]).

Although a solution is not generally available in closed-form, one can again exploit the similarity between the expression for h_* in (8) and the total cost for an associated fluid model control problem. It is known that h_* is approximated by the optimal total cost for the fluid model, and that the latter is piecewise quadratic whenever c is piecewise linear (see [47], and Section 3.1 below). The piecewise linear assumption on c captures a wide variety of cost functions including, for example, the ℓ_1 ($c(q) = \sum_i q_i$) and ℓ_∞ ($c(q) = \max_i q_i$) norms.

However, the reduction to a fluid model with no variability still leaves substantial effort to find an optimal or approximately optimal policy for the stochastic model. In particular, even the fluid model optimal control problem is intractable for network models with numerous buffers and resources. The construction of effective policies begins with the workload relaxation for the fluid model to obtain an optimization problem of lower dimension. This construction ends with methods of translating a policy from the fluid to the stochastic domain.

For details we refer the reader to [49, 12, 45]. Here we restrict attention to the KSRS model for the sake of example. The construction of an effective policy for this example involves the following steps:

Step 1: Consideration of the fluid model The workload relaxation for this model is the differential inclusion $\frac{d}{dt}\widehat{q} \in \widehat{\mathsf{V}}$ where in this two-dimensional relaxation,

$$\widehat{\mathsf{V}} = \left\{v : \langle \xi^i, v\rangle \geq -(1-\rho_i),\ i = 1, 2\right\}, \qquad \xi^1 = \begin{bmatrix} m_1 \\ 0 \\ m_4 \\ m_4 \end{bmatrix},\ \xi^2 = \begin{bmatrix} m_2 \\ m_2 \\ m_3 \\ 0 \end{bmatrix},$$

and $\rho_i := \langle \xi^i, \alpha\rangle$ is the load at Station i for $i = 1, 2$. This can be equivalently expressed as the fluid model $\frac{d}{dt}\widehat{q}(t) = B\widehat{\zeta}(t) + \alpha$, where $(\widehat{\boldsymbol{q}}, \widehat{\boldsymbol{\zeta}})$ are subject to the constraints,

$$\widehat{q}(t) \in \mathbb{R}^4_+, \qquad \widehat{\zeta}(t) \in \mathbb{R}^4,\ C\widehat{\zeta}(t) \leq \boldsymbol{1}\ .$$

That is, $\widehat{\boldsymbol{q}}$ is precisely the fluid model (5), except that the positivity constraints on $\boldsymbol{\zeta}$ have been relaxed.

The set of feasible workload values is given by $\widehat{\mathsf{W}} := \{\widehat{\Xi}x : x \in \mathsf{X}\}$, with $\widehat{\Xi} = [\xi^1 \mid \xi^2]^{\scriptscriptstyle T}$. The workload space for the KSRS model is equal to $\mathbb{R}^2_+$ if we assume that $\mathsf{X} = \mathbb{R}^4_+$ (infinite waiting-room).

Two states $x, y \in \mathsf{X}$ are called *exchangable* if they have identical workload values $\langle x, \xi^i\rangle = \langle y, \xi^i\rangle$ for $i = 1, \ldots, n$. In this case, the minimal time to reach x from y is *zero* in the two-dimensional relaxation. We can therefore view a workload relaxation as decomposing the possible buffer states into equivalence classes. In identifying optimal paths in this setting, one need only concentrate on paths through the equivalence classes, since one always moves instantaneously to a set of buffer levels that has minimal cost within the current equivalence class.

For a given vector $w \in \widehat{\mathsf{W}}$, the *effective cost* $\bar{c}(w)$ is defined to be the cost of the 'cheapest' state with this workload value. When the cost function c is linear, the effective cost is the solution to the linear program,

$$\begin{aligned} \bar{c}(w) = \quad & \mathbf{min} \quad \langle c, x \rangle \\ & \mathbf{subject\ to} \quad \begin{aligned} \widehat{\Xi} x &= w \\ x &\in \mathsf{X}. \end{aligned} \end{aligned} \tag{9}$$

When $\mathsf{X} = \mathbb{R}^4_+$ the dual may be written,

$$\begin{aligned} \bar{c}(w) = \quad & \mathbf{max} \quad \langle \gamma, w \rangle \\ & \mathbf{subject\ to} \quad \begin{aligned} & \widehat{\Xi}^T \gamma \le c \\ & \gamma \textit{ not sign-constrained} \end{aligned} \end{aligned} \tag{10}$$

Using this form we can write $\bar{c}(w) = \max_i(\langle \bar{c}^i, w \rangle)$, $w \in \widehat{\mathsf{W}}$, where $\{\bar{c}^i\}$ are the extreme-points in the constraint region in (10).

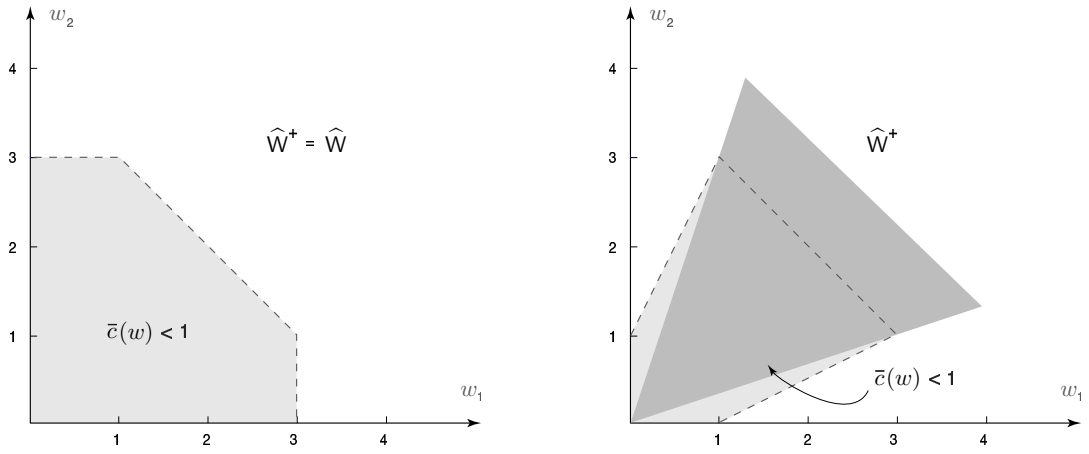

Figure 3: Level sets of the effective cost for the KSRS model in Cases I and II respectively.

The effective cost is shown in Figure 3 for the case where $\mathsf{X} = \mathbb{R}^4_+$, and c is the ℓ_1-norm. We consider two cases:

Case I $\mu_2 = \mu_4 = 1/3$, $\mu_1 = \mu_3 = 1$, and $\alpha_1 = \alpha_3 < 1/4$. In this case $\{\bar{c}^i : i = 1, 2, 3\}$ are given by,

$$\left\{ \tfrac{1}{3} \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \ \tfrac{1}{4} \begin{pmatrix} 1 \\ 1 \end{pmatrix} , \ \tfrac{1}{3} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and the loads $\{\rho_i\}$ at each station are equal to the system load $\rho = 4\alpha_1$.

Case II $\mu_2 = \mu_4 = 1$, $\mu_1 = \mu_3 = 1/3$, and $\alpha_1 = \alpha_3 < 1/4$. In this case $\{\bar{c}^i : i = 1, 2, 3\}$ are given by,

$$\left\{ \begin{pmatrix} 1 \\ -2 \end{pmatrix} , \ \tfrac{1}{4} \begin{pmatrix} 1 \\ 1 \end{pmatrix} , \ \begin{pmatrix} -2 \\ 1 \end{pmatrix} \right\}$$

and the loads at each station are again equal to $\rho = 4\alpha_1$.

The region on which $\bar{c}(\,\cdot\,)$ is monotone plays an important role in identifying optimal policies. This is given by,

$$\widehat{\mathsf{W}}^+ := \{w \in \widehat{\mathsf{W}} : \bar{c}(w) \le \bar{c}(w') \quad \text{whenever } w' \ge w\}.$$

We say that $\bar{c}$ is *monotone* when $\widehat{\mathsf{W}}^+ = \widehat{\mathsf{W}}$. Monotonicity frequently fails to hold since, in many models, one must starve a resource (and therefore increase work for that resource) to remove material from the system. In Case I we have $\widehat{\mathsf{W}}^+ = \widehat{\mathsf{W}}$ (the effective cost is monotone), and in Case II the monotone region $\widehat{\mathsf{W}}^+$ is a strict subset as shown in Figure 3.

Figure 4 illustrates the effective cost for a more complex manufacturing model considered in [17]. It is highly non-monotone since the topology of this network imposes starvation of resources to drain material from the network. For example, to meet demand d_1 it may be necessary to temporarily starve Station 1.

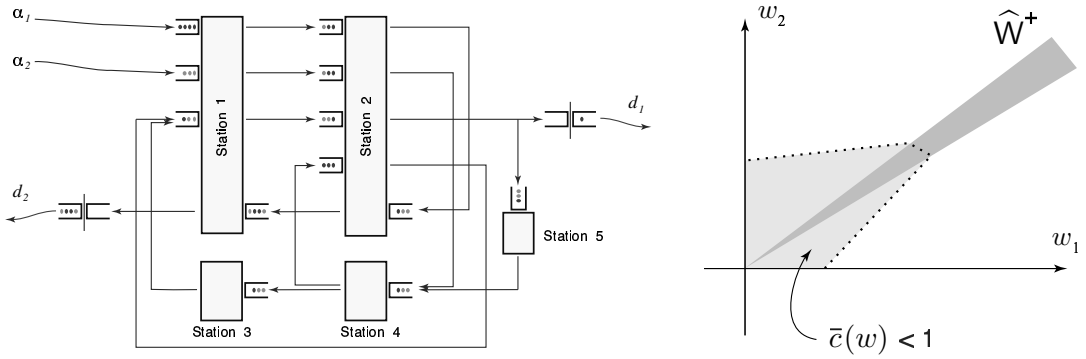

Figure 4: A manufacturing model and its effective cost. This network has 16 buffers, four of which are *virtual*, corresponding to backlog or excess inventory. There are two bottlenecks, so a two-dimensional relaxation is again appropriate for control design.

Once the effective cost has been identified we then find a policy that minimizes the total cost (11) over all policies,

$$\widehat{J}(w) := \int_0^\infty \bar{c}(\widehat{w}(t))\, dt, \qquad \widehat{w}(0) = w\,. \tag{11}$$

The optimal solution is *path-wise optimal* in each case for the KSRS example. In Case II it satisfies $\widehat{w}^*(t) \in \widehat{\mathsf{W}}^+$ for all $t > 0$ even if $\widehat{w}(0) \notin \widehat{\mathsf{W}}^+$, since in this case the relaxation permits an instantaneous increase in one of the workload elements to bring the workload onto the boundary of the set $\widehat{\mathsf{W}}^+$.

Step 2: The impact of variability In general, once we obtain a policy in workload space for the fluid model we must compensate for variability to translate to the original network. An optimal solution for the fluid workload relaxation is described by the *linear policy*: Resource 2 is required to work at maximal rate if $w_2 > s_1^* w_1$, and resource 1 must work at maximal rate if $w_1 > s_2^* w_2$. In Case I we have $s_1^* = s_2^* = 0$, and in Case II we

have $s_1^* = s_2^* = 1/3$. That is, the switching curves defining the optimal policy are simply the boundaries of the monotone region $\widehat{\mathsf{W}}^+$.

For a stochastic model we consider *affine* policies of the specific form,

$$\begin{aligned} &\textit{Work resource 2 at maximal rate if} \quad w_2 > s_1^* w_1 - \beta_1; \\ &\textit{Work resource 1 at maximal rate if} \quad w_1 > s_2^* w_2 - \beta_2. \end{aligned} \tag{12}$$

In Cases I and II one can construct a policy based upon an affine switching curve in workload space such that the mean $\mathsf{E}[\bar{c}(\widehat{W}(t))]$ is approximately optimized *for all* $t > 0$ in a Brownian workload relaxation. The error vanishes exponentially quickly with t. A formula for the optimal affine policy was obtained in [12]. In Case I the optimal policy for the workload relaxation is non-idling regardless of variability (i.e. $s_i^* = \beta_i^* = 0$). In Case II, β^* grows linearly with increasing variability.

Step 3: Translation to 'buffer coordinates' Once an optimal policy is identified for a relaxed model one may translate back to to the original discrete model by exploiting 'near-exchangeability' of states with identical effective workload. There are several issues that must be addressed to make such translation possible:

(i) The discrete network model presents constraints that do not exist in the fluid or stochastic workload relaxations. For example, in the fluid model shown in Figure 1, it is possible for Station 2 to be empty ($q_2(t) = q_3(t) = 0$), yet to have full utilization at that machine: $\zeta_2(t) + \zeta_3(t) = 1$. This is not possible for the discrete network, but starvation can be avoided by imposing safety-stocks at various buffers so that no critical resource is starved of work. The possibility of starvation of critical resources is typically taken care of by the control policies in a pure routing control problem. This is not true in scheduling.

(ii) Another issue in translation is that the allocation rates ζ lie in a general polyhedron U. For the discrete network it may not be physically meaningful to allow continuous values for $U(k)$ in a 2-parameter model, or for $\frac{d}{dt}Z(t)$ in the general model (1).

(iii) Finally, any translation of a fluid allocation must concentrate a resource to work on groups of parts consecutively to avoid excessive switch-overs. For example, in the manufacture of semiconductors, chips are processed in lots.

Many of these issues are addressed by interpreting ζ appropriately. The allocation rates determined by an analysis of the fluid model should not be strictly interpreted. Given a rate ζ and a time $[T_1, T_2]$ for control, the allocation $\boldsymbol{Z}$ for the physical network should provide $\zeta_i(T_2 - T_1)$ seconds of service to activity i, $1 \le i \le \ell_a$, while maintaining other physical constraints. For example, we may require that the entries of $\frac{d}{dt}Z(t)$ take on binary values between control switch-overs [23, 43, 44].

To illustrate the value of safety-stocks consider again the KSRS model. For the Brownian model the optimal policy is determined by two switching curves that are approximately affine. For the 2-parameter model we impose the constraint that Station 2 cannot idle if

$$W_2 \ge s_i^* W_1 - \beta_1^* \,,$$

with $W_i := \langle \xi^i, Q \rangle$. A similar condition of course is given for Station 1. When the above condition holds then one must enforce a safety-stock condition whenever Station 2 approaches starvation.

Conditioned on $Q(0) = x$, the 'time to starvation' at Station 2 is given by $T_2 := \min(t : Q_2(t) + Q_3(t) = 0)$. A lower bound on its mean for the 2-parameter model is given by,

$$\mathsf{E}[T_2 \mid Q(0) = x] \geq m_2 x_2 + m_3 x_3.$$

Based on this observation, and the form of the optimized workload relaxation, we are motivated to consider the following,

Affine policy for the KSRS model
Serve buffer 1 if buffer 4 is empty, or

$$W_2 \geq s_i^* W_1 - \beta_1^* \quad \textit{and} \quad m_2 Q_2 + m_3 Q_3 \leq \bar{w}_2. \tag{13}$$

An analogous condition holds at Station 2.

This policy is approximately optimal in heavy traffic provided the safety-stock parameters $\{\bar{w}_i\}$ are chosen of order $|\log(1-\rho)|$ [49, 12]. However, sensitivity to $\{\bar{w}_i\}$ may be large for systems with moderate load.

3 Variance Reduction in Simulation

The methods described in the previous section give rise to a family of affine policies, together with safety-stock levels that must be chosen based on variability and structure of the given network. We have already remarked that network performance can be sensitive to the choice of safety-stock levels, and the translation of a policy for the fluid network to one for the discrete network will almost invariably involve a search for appropriate safety-stock levels.

3.1 Approaches to performance approximation

We begin with the problem of determining the expected cost of a given policy. One approach is to exploit the Markovian structure of the 2-parameter model. The state process $\boldsymbol{Q}$ is a time-homogeneous Markov chain whenever $\boldsymbol{U}$ is determined by state-feedback, so that $U(k) = \phi(Q(k))$, $k \geq 0$, for some fixed function $\phi \colon \mathsf{X} \to \mathsf{U}$. Under mild conditions, the Markov chain $\boldsymbol{Q}$ will be positive recurrent, and hence steady-state cost is determined by an invariant probability distribution π on X.

Consider for example the 2-parameter workload model in one dimension given by

$$\widehat{W}(k+1) = \widehat{W}(k) - 1 + I(k) + D(k+1), \qquad k \geq 0\,, \tag{14}$$

where $\boldsymbol{D}$ is an i.i.d. sequence supported on $\mathbb{Z}_+$, with common mean $\mathsf{E}[D(k)] = \rho < 1$, and finite variance. We restrict $\boldsymbol{W}$ to the state space $\widehat{\mathsf{W}} = \mathbb{Z}_+$ to simplify calculation. The sequence $\boldsymbol{I}$ is the idleness process which is assumed to take values in $\mathbb{Z}_+$. Under the

assumption that $\boldsymbol{I}$ is chosen to be non-idling, $I(k) = \mathbb{I}(\widehat{W}(k) = 0)$, we can compute the steady-state mean,

$$\gamma := \lim_{k\to\infty} \mathsf{E}[\widehat{W}(k)] = \sum_{w\in\widehat{\mathsf{W}}} w\pi(w) = \tfrac{1}{2}\frac{\sigma_D^2}{1-\rho} + \frac{\rho}{2}.$$

Direct calculation shows that the function $h\colon \widehat{\mathsf{W}} \to \mathbb{R}$ given below is the associated *relative value function*,

$$h(w) = \tfrac{1}{2}\frac{1}{1-\rho}\Big(w^2 + (1-2\rho)w\Big)\,. \tag{15}$$

That is, h solves *Poisson's equation*,

$$\mathsf{E}[h(\widehat{W}(k+1)) \mid \widehat{W}(k) = w] = h(w) - w + \gamma\,. \tag{16}$$

The relative value function may be expressed as the expectation,

$$h(w) = \mathsf{E}\Big[\sum_{k=0}^{\tau_\theta - 1} [\bar{c}(\widehat{W}(k)) - \gamma] \mid \widehat{W}(0) = w\Big]\,, \tag{17}$$

where $\bar{c}(w) = w$ (see also Section 3.2).

The continuous-time Brownian workload model can be similarly analysed in one dimension. For the non-idling policy we have

$$h(w) = \mathsf{E}\Big[\int_{t=0}^{\tau_\theta} [\bar{c}(\widehat{W}(t)) - \gamma]\, dt \mid \widehat{W}(0) = w\Big] = \tfrac{1}{2}\frac{1}{1-\rho}w^2 \tag{18}$$

where again $\bar{c}(w) = w$, and in this case the steady-state cost is given by,

$$\gamma = \lim_{t\to\infty} \mathsf{E}[\widehat{W}(t)] = \tfrac{1}{2}\frac{\sigma_N^2}{1-\rho}$$

with σ_N^2 the instantaneous covariance of the Brownian motion $\boldsymbol{N}$ given in (6). We see that the steady-state mean for the 2-parameter model and its Brownian approximation are in nearly perfect agreement for ρ near unity provided $\sigma_D^2 = \sigma_N^2$.

Note that the steady-state mean γ does not depend upon the precise distribution of the random input, but only on its first and second moments for both the Brownian and 2-parameter workload models. Also, the relative value function is *independent of variability* for both the 2-parameter and Brownian workload models. In the special case where there is no variability in (6) we obtain the one-dimensional fluid model, $\frac{d}{dt}\widehat{w} = -1 + \iota + \rho$, with $\boldsymbol{\iota}$ the non-idling policy: $\iota(t) = 0$ if $\widehat{w}(t) > 0$. With $\bar{c}(w) \equiv w$ the total cost (11) is the quadratic,

$$\widehat{J}(w) = \tfrac{1}{2}\frac{1}{1-\rho}w^2\,.$$

This is precisely equal to the relative value function for the Brownian model, and the quadratic part of the relative value function for the 2-parameter model.

This solidarity is a special case of the approximations obtained in [46, 47, 48, 12]. The relative value function for a stochastic queueing model is approximated by a fluid value

function under general conditions for general network models. The key assumption is some form of stability so that the expression (17) is well defined.

Consider now the two-dimensional case. In this case we cannot explicitly solve Poisson's equation or compute the steady-state cost in general. However, we can use the previous calculations to obtain some insight.

For the two-dimensional Brownian workload model on $\widehat{\mathsf{W}} = \mathbb{R}_+^2$ the solution to Poisson's equation is again given by the left hand side of (18), with γ equal to the steady-state mean of $\bar{c}(\widehat{W}(t))$. Let $\kappa\Sigma$ denote the instantaneous covariance of the Brownian motion $\boldsymbol{N}$ in (6), where $\Sigma > 0$ is a 2×2 matrix, and $\kappa \geq 0$ is a scaling-parameter. Let $\gamma(\kappa)$ denote the steady-state cost for a given κ, and $h(w;\kappa)$ the relative value function. With the cost function fixed, and the policy also fixed as the non-idling policy, the steady-state cost scales linearly with κ so that $\gamma(\kappa) = \kappa\gamma$ with $\gamma{:=}\gamma(1)$. We also have $h(w;\kappa) = \kappa^2 h(w/\kappa;1)$ for $\kappa > 0$, and $h(w;\kappa) \to \widehat{J}(w)$ as $\kappa \to 0$, where $\widehat{J}$ is again the total cost for the fluid model [12].

Suppose that the relative value function is sufficiently smooth in κ, so that for some function $\widehat{S}\colon \widehat{\mathsf{W}} \times \mathbb{R}_+ \to \mathbb{R}$, we have the first-order Taylor series expansion,

$$h(w;\kappa) = \widehat{J}(w) + \kappa\widehat{S}(w) + O(\kappa^2), \quad w \in \widehat{\mathsf{W}}.$$

Given the scaling properties of h it is reasonable to assume that this bound holds with error term uniformly bounded in w (see [12, Proposition 3.5]). We obtain a second formula on noting that $h(\,\cdot\,;\kappa)$ solves a partial differential equation for any positive κ. Define the differential operator $\mathcal{D}$ via

$$\mathcal{D}h := -\langle \boldsymbol{1} - \boldsymbol{\rho}, \nabla h\rangle + \frac{\kappa}{2}\mathrm{trace}\,(\Sigma\Delta h)\,. \tag{19}$$

Recall that h solves Poisson's equation, so that for each fixed κ

$$\begin{aligned} -\bar{c}(w) + \gamma(\kappa) &= \mathcal{D}h(w;\kappa) \\ &\approx \; -\langle \boldsymbol{1} - \boldsymbol{\rho}, \nabla\widehat{J}(w)\rangle + \frac{\kappa}{2}\mathrm{trace}\,(\Sigma\Delta\widehat{J}) - \langle \boldsymbol{1} - \boldsymbol{\rho}, \nabla\widehat{S}(w)\rangle + \frac{\kappa^2}{2}\mathrm{trace}\,(\Sigma\Delta\widehat{S}) \end{aligned} \tag{20}$$

Note that $\langle \boldsymbol{1} - \boldsymbol{\rho}, \nabla\widehat{J}(w)\rangle = \bar{c}(w)$. Differentiating both sides of (20) with respect to κ and setting $\kappa = 0$ we obtain the ODE

$$-\langle \boldsymbol{1} - \boldsymbol{\rho}, \nabla\widehat{S}(w)\rangle = \gamma - \tfrac{1}{2}\mathrm{trace}\,(\Sigma\Delta\widehat{J}(w))\,.$$

We also have the boundary condition $\widehat{S}(\boldsymbol{\theta}) = 0$ since $\widehat{J}(\boldsymbol{\theta}) = h(\boldsymbol{\theta};\kappa) = 0$ for all $\kappa \geq 0$. If we assume moreover that $\widehat{S}(\,\cdot\,)$ is a continuous function on $\widehat{\mathsf{W}}$ we can solve this ODE to obtain,

$$\widehat{S}(w) = \int_0^{\tau_\theta} [\tfrac{1}{2}\mathrm{trace}\,(\Sigma\Delta\widehat{J}(\widehat{w}(t))) - \gamma]\,dt$$

where in the right hand side we initialize $\widehat{w}(0) = w$, and $\tau_\theta = \max_{i=1,2} w_i(1-\rho_i)^{-1}$ is the emptying time for the fluid model.

In conclusion, we find that under the assumed smoothness conditions the Taylor series approximation holds,

$$h(w;\kappa) \approx \widehat{J}(w) + \kappa\widehat{S}(w)\,. \tag{21}$$

The correction term $\widehat{S}$ is piecewise linear in w provided $\bar{c}$ is piecewise linear, since in this case $\widehat{J}$ is piecewise quadratic and hence $\Sigma\Delta\widehat{J}(\widehat{w}(t))$ is piecewise constant in t.

These calculations are based on a non-idling policy. Similar approximations hold for affine policies, provided we assume that $\{\beta_i\}$ scale linearly with κ in (12).

This leads to several approaches to performance evaluation. The fact that the relative value function h for the 2-parameter model (3) is roughly quadratic in shape motivates quadratic or piecewise quadratic approximations, as in the treatment of generalized Jackson networks in [50], or in treatments of simple queues [51].

Linear programs are constructed in the search for a quadratic approximation for general network models in [35, 6, 34, 52]. This gives simultaneously bounds on performance and criteria for stability. These approaches are based on the assumption that the primitive processes in (1) are Poisson processes, although the method extends easily to 2-parameter models on an integer lattice. Extensions of these ideas to Brownian models are presented in [58].

Unfortunately, the linear programming technique cannot be easily applied in our context because of the potential complexity of the operating policy. Furthermore, in order to distinguish between performance for similar policies with different safety-stock levels one would need to add constraints specific to each safety-stock level in the associated linear program. It is not clear how to introduce such constraints in the linear programming approaches obtained to-date. Finally, there is no guarantee that the resulting bounds on performance for different safety-stock levels will enable us to distinguish between policies.

We are thus led to the use of simulation techniques to evaluate performance. This approach is of course flexible, and we have seen in related work that the structure of typical policies for fluid network models lends itself to construction of reduced-variance simulators [26, 27, 24, 25]. We discuss extensions of these techniques next.

3.2 Shadow functions

We begin with some observations regarding simulation of sample means, following ideas from [26, 27, 24, 25]. We consider a positive recurrent Markov chain $\boldsymbol{Y} = (Y(m) : m \geq 0)$ with state space X, and let $f\colon \mathsf{X} \to \mathbb{R}$ denote some measurable function. Our goal is to compute $\bar{f} := \pi(f)$, the steady state mean, with π the invariant distribution for the chain.

For a given *forcing function* $f\colon \mathsf{X} \to \mathbb{R}$, *Poisson's equation* in this general framework takes the form,

$$Ph = h - f + \bar{f}, \tag{22}$$

where $Ph(x) = E[h(Y(1))|Y(0) = x]$. The relative value function h plays a central role in the analysis of simulation algorithms, just as it does in the theory of Markov decision processes (recall the dynamic programming equations (8)). While it is not uniquely defined (for example, one may add a constant to obtain a new solution), there is a generic choice involving the *fundamental kernel* Z. Let $P^m f(x) = E[f(Y(m))|Y(0) = x]$, $m \geq 0$. For any function $f\colon \mathsf{X} \to \mathbb{R}$ the fundamental kernel is given by

$$h(x) = Zf\,(x) = \sum_{m=0}^{\infty} [-\pi(f) + P^m f\,(x)] \tag{23}$$

Of course, some conditions must be imposed to ensure that these sums are well-defined.

A convenient and unrestrictive sufficient condition is geometric ergodicity. It is known that most definitions of geometric ergodicity are equivalent to the following uniform formulation [31, 30, 51]. For a given function $V\colon \mathsf{X} \to [1,\infty)$, the function space L_∞^V is equal to the set of all Borel-measurable functions $\psi : \mathsf{X} \to \mathbb{R}$ satisfying

$$\|\psi\|_V := \sup_{x\in\mathsf{X}} \frac{|\psi(x)|}{V(x)} < \infty.$$

The Markov chain is called *V-uniformly ergodic* if there is a unique invariant distribution π, and P^n converges to π in the induced-operator norm on L_∞^V. Equivalently, there exist constants $K < \infty, \delta > 0$ satisfying

$$\|P^n\psi - \pi(\psi)\|_V := \sup_{x\in\mathsf{X}}\Big(\frac{|P^n\psi\,(x) - \pi(\psi)|}{V(x)}\Big) \le K\|\psi\|_V e^{-\delta n} \qquad n \ge 0,\ \psi \in L_\infty^V.$$

See [51, Chapter 16], or [33] for recent refinements. Throughout this section we assume that $\boldsymbol{Y}$ is V-uniformly ergodic, and that $f^4 \in L_\infty^V$.

The standard Monte-Carlo estimator of $\bar{f}$ is given by

$$\bar{f}(n) := \frac{1}{n}\sum_{m=0}^{n-1} f(Y(m)), \quad n \ge 1\,, \tag{24}$$

which is always strongly consistent under the assumptions imposed here. However, in the applications considered below we typically have a great deal of prior information that can be used to construct much more efficient estimators. Fix one measurable function satisfying $g^2 \in L_\infty^V$, $\pi(g) = 0$, and consider for any $r \in \mathbb{R}$ the *smoothed estimator*,

$$\bar{f}_s(n) := \frac{1}{n}\sum_{m=0}^{n-1} [f(Y(m)) - g(Y(m))]\,, \quad n \ge 1.$$

Under V-uniform ergodicity one can infer that the smoothed estimator is strongly consistent, i.e., $\bar{f}_s(n) \to \bar{f}$, as $n \to \infty$.

But how can we construct functions with the required zero-mean property? A rich class is obtained as follows (see [25, 29]). Take any function h satisfying $h^2 \in L_\infty^V$, and set

$$g = h - Ph\,.$$

Then $g \in L_\infty^V$, and satisfies $\pi(g) = 0$. Based on this definition we find that h is the solution to Poisson's equation with forcing function g. We call functions g of the form $h - Ph$ *shadow functions*.

An optimal choice is obtained when $h = Zf$, the solution to Poisson's equation (22) with forcing function equal to f. In this case, the estimator has zero variance: $\bar{f}_s(n) = \bar{f}$ almost surely. While the function h is not computable in general, this observation provides a rich class of functions g for which $\pi(g) = 0$, and suggests that a good choice for h is some function that nearly solves Poisson's equation. Recall that approximations for $h = Zf$ were constructed for network models in Section 3.1 based on the fluid model. Let us see whether these approximations can yield improvements in accuracy in simulations of 2-parameter models.

3.3 Variance reduction for the KSRS model

Consider the KSRS model shown in Figure 1. We give results only for Case I, in which the output buffers 2 and 4 are slower than the input buffers 1 and 3, since the results in Case II are similar. For this example we take as cost function c the ℓ_1 norm, i.e., we estimate the expected steady-state number of jobs in the system.

Let $\boldsymbol{Y} \equiv \boldsymbol{Q}$ denote the 2-parameter model for this network. We take $\mu = [1, 1/3, 1, 1/3]$ and $\alpha = 0.9[1/4, 0, 1/4, 0]$, scaled so that the components of μ and α sum to 1. We mimic Poisson statistics by taking the increment distributions to all be of the form (4) with $\eta = 1$ and κ equal to the inverse of the rates (for each of the 6 primitive processes). We use the policy given in (13) with safety stock levels $\bar{w}_i = 50$ for $i = 1, 2$.

As we will see, the standard estimator (24) requires very large simulation runlengths to converge, and so we are motivated to explore variance reduction using shadow functions. For this model the solution to Poisson's equation, denoted $h := Zc$, is approximated by the value function $\widehat{J}$ defined in (11): Letting $h(y; 0) := \widehat{J}(\Xi y)$, we have $h(y; 0) \approx h(y)$, $y \in \mathsf{X}$. This motivates the shadow function

$$g(y) = h(y; 0) - \mathsf{E}[h(Y(1); 0)|Y(0) = y], \tag{25}$$

which in turn yields the *smoothed estimator*

$$\bar{f}_s(n) := \frac{1}{n} \sum_{m=0}^{n-1} [f(Y(m)) - g(Y(m))], \quad n \geq 1. \tag{26}$$

The results are shown in Figure 5. Notice that the fluctuations of the standard estimator are tremendous while those of the smoothed estimator are almost nonexistent. This translates into tremendous variance reduction. Indeed, by repeating this simulation experiment 50 independent times we find that the smoothed estimator reduces the variance of the standard estimator by a factor of more than 100.

The smoothed estimator above differs from those given in [26, 27]. In [26] we considered shadow functions of the form $h - Ph$, where h was a (pure) quadratic function of the buffer levels. The function h was chosen by solving a linear program. This differs from the present case in that we consider here a piecewise, not pure, quadratic function, and the argument of the function is equal to a vector of workload values, rather than buffer levels. Large variance reductions were obtained in [26] for networks in moderate traffic, but in heavy traffic the variance reduction factors were not as large, perhaps because the pure quadratic was not able to capture the true form of the solution to Poisson's equation.

This work was extended in [27] to piecewise quadratic functions of the buffer levels, following numerical results obtained in [13]. In principle, this class of functions subsumes the piecewise quadratic functions of workloads considered here, so that one would expect larger variance reductions to be possible in that setting. The advantage of the current approach is that we can compute an approximation to the fluid value function directly from the workload relaxation. Hence, we can exploit our understanding of workload relaxation to identify suitable shadow functions, even in networks of moderate to significant complexity, such as that depicted in Figure 4.

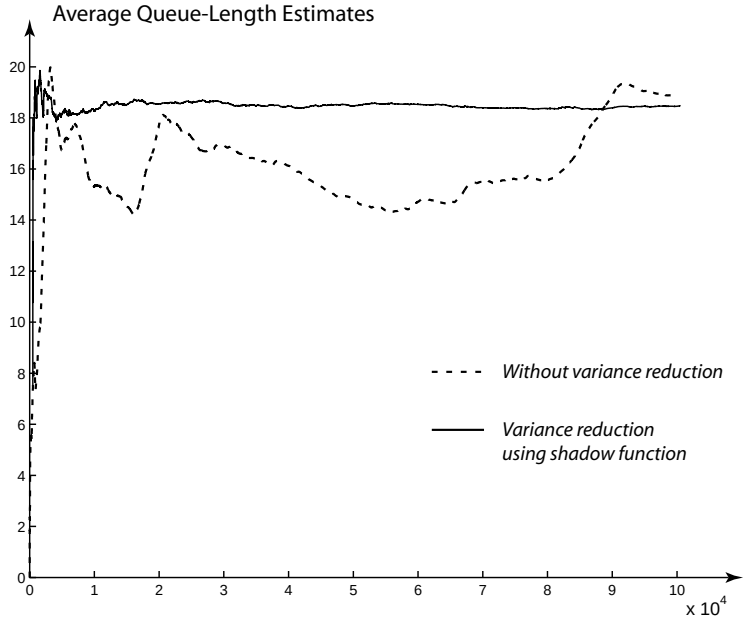

Figure 5: Results for a simulation run of length 100,000 steps. The dashed line represents the running average cost using the standard estimator $\bar{f}(n)$ defined in (24). The solid line represents the running average cost for the estimator (26).

3.4 A general simulation algorithm

So far, the simulation approaches we have described are based on a single shadow function. In practice it is likely that we may construct a family of shadow functions, indexed by a multi-dimensional parameter r. We consider here approaches to dynamically search for a good choice of r in this general setting.

We first consider a specialized context to fix ideas, and relate these ideas to previous work. Fix one measurable function satisfying $g^2 \in L_\infty^V$, $\pi(g) = 0$, and consider for any $r \in \mathbb{R}$ the estimator,

$$\bar{f}(n;r) := \frac{1}{n} \sum_{m=0}^{n-1} [f(Y(m)) - rg(Y(m))] \,.$$

Under V-uniform ergodicity one can again infer that this estimator is strongly consistent, i.e., $\bar{f}(n;r) \to \bar{f}$ as $n \to \infty$, for any r. But how should r be chosen?

For example, suppose we choose to minimize the steady-state variance. A simple calculation gives the optimal value,

$$r_0^* := \arg\min \big\{ \mathrm{Var}\left(f(Y(m)) - rg(Y(m))\right) : r \in \mathbb{R} \big\} = \frac{\langle f, g \rangle_\pi}{\langle g, g \rangle_\pi},$$

where for square-integrable functions $\{f_1, f_2\}$ we define $\langle f_1, f_2 \rangle_\pi := \mathsf{E}_\pi[f_1(Y(k)) f_2(Y(k))]$ (independent of $k \geq 0$). However, in simulation one is interested in the variance of the estimator $\bar{f}(n;r)$, and not the variance of the summands used to define this estimator.

Maintaining the assumption that $g^2 \in L_\infty^V$, it may be shown that for any $r \in \mathbb{R}$, the normalized estimates

$$n^{\frac{1}{2}}(\bar{f}(n;r) - \bar{f}), \qquad n \geq 1, \tag{27}$$

are approximately Gaussian. The asymptotic mean is zero, and the asymptotic variance is given by

$$s(r) := \lim_{n\to\infty} n\mathsf{E}_\pi[(\bar{f}(n;r) - \bar{f})^2] \tag{28}$$

The quantity (28) is also known as the time-average variance constant (TAVC), and has several equivalent representations (see [51, Chapter 17]). One that is useful here is expressed in terms of Poisson's equation via,

$$s(r) = \langle\!\langle Z(f - rg), Z(f - rg) \rangle\!\rangle_\pi, \tag{29}$$

where the fundamental kernel Z is defined in (23), and for any square-integrable functions f_1, f_2,

$$\langle\!\langle f_1, f_2 \rangle\!\rangle_\pi := \pi\big((f_1 f_2) - (Pf_1)(Pf_2)\big)\,.$$

Given the expression (29) we obtain the following optimal value,

$$r^* = \frac{\langle\!\langle Zg, Zf \rangle\!\rangle_\pi}{\langle\!\langle Zg, Zg \rangle\!\rangle_\pi},$$

which is analogous to the expression for the value r_0^* minimizing the steady-state variance of $f - rg$.

This then is the well-known variance reduction technique of control variates, as specialized to steady-state simulation of V-uniformly ergodic Markov chains. Control variates may be viewed as *linear* parameterizations of shadow functions. We next consider nonlinear parameterizations.

Suppose that $g\colon \mathsf{X} \times \mathbb{R}^p \to \mathbb{R}$ is C^1 in its second variable, and continuous as a function of two variables. For any fixed $r \in \mathbb{R}^p$ we may consider the estimator,

$$\bar{f}(n;r) := \frac{1}{n}\sum_{m=0}^{n-1} [f(Y(m)) - g(Y(m);r)] \tag{30}$$

Note that the parameterization is now smooth, rather than linear, and we are now considering multivariate parameterizations.

It was shown in [20] that nonlinear parameterizations possess no asymptotic advantage over linear parameterizations. However, the parameterization (30) is *not* of the form considered in [20], so this negative result does not apply. Indeed, the parameterization considered in [20], as viewed in our context, would be of the form

$$\frac{1}{n}\sum_{m=0}^{n-1} f(Y(m)) - g\left(\frac{1}{n}\sum_{m=1}^{n} Y(m); r\right).$$

As in the previous section, we assume that, for any $r \in \mathbb{R}^p$, the function $g(x;r)$ satisfies,

$$\int g(x;r)\pi(dx) = 0, \tag{31}$$

so that (30) is a consistent estimator of $\bar{f}$. In the applications considered below we take g of the form

$$g(x;r) = h(x;r) - \int P(x,dy)h(y;r), \qquad x \in \mathsf{X},\ r \in \mathbb{R}^p\,, \tag{32}$$

where $h\colon \mathsf{X} \times \mathbb{R}^p$ satisfies appropriate smoothness properties.

For a given function g satisfying (31), our goal is to find $r \in \mathbb{R}^p$ that minimizes the TAVC $s(r)$. However, due to numerical considerations we consider the following discounted approximation. For a fixed $0 \le \lambda \le 1$ define,

$$\begin{aligned} s_\lambda(r) \;\; := \;\; & 2\sum_{n=0}^{\infty} \lambda^n \mathsf{E}_\pi[(f(Y(0)) - g(Y(0);r) - \bar{f})(f(Y(n)) - g(Y(n);r) - \bar{f})] \\ & -\mathsf{E}_\pi[(f(Y(0)) - g(Y(0);r) - \bar{f})^2] \end{aligned} \tag{33}$$

In (33) we assume that the Markov chain $\boldsymbol{Y} = \{Y(k) : -\infty < k < \infty\}$ is stationary, with common marginal distribution π. Under Assumptions A1-A4 below we have $s_\lambda(r) \to s_1(r)$ as $\lambda \uparrow 1$ (note that $s(r) = s_1(r)$).

To find a value of r minimizing s_λ we require an expression for its gradient, $\nabla s_\lambda(\cdot)$, with respect to r. Differentiating (33) gives the following representation:

$$\begin{aligned} \nabla s_\lambda(r) \;\; = \;\; & 2\mathsf{E}_\pi[\nabla_r g(Y(0);r)(f(Y(0)) - g(Y(0);r))] \\ & -2\textstyle\sum_{n=0}^{\infty} \lambda^n \mathsf{E}_\pi[\nabla_r g(Y(0);r)(f(Y(n)) - g(Y(n);r))] \\ & -2\textstyle\sum_{n=0}^{\infty} \lambda^n \mathsf{E}_\pi[\nabla_r g(Y(n);r)(f(Y(0)) - g(Y(0);r))]. \end{aligned} \tag{34}$$

We have used the fact that $\mathsf{E}_\pi \nabla_r g(Y(n);r) = 0$ for all $n \ge 0$ as follows from (31).

Under the assumption that $\mathbf{Y}$ is stationary, (34) can be rewritten as

$$\begin{aligned} \nabla s_\lambda(r) \;\; = \;\; & 2\mathsf{E}_\pi[\nabla_r g(Y(n);r)(f(Y(n)) - g(Y(n);r))] \\ & -2\textstyle\sum_{i=-\infty}^{n} \lambda^{n-i} \mathsf{E}_\pi[\nabla_r g(Y(i);r)(f(Y(n)) - g(Y(n);r))] \\ & -2\textstyle\sum_{i=-\infty}^{n} \lambda^{n-i} \mathsf{E}_\pi[\nabla_r g(Y(n);r)(f(Y(i)) - g(Y(i);r))], \qquad n \ge 0, \end{aligned}$$

and this suggests the estimator

$$\begin{aligned} \widehat{\nabla} s_\lambda^n(r) \;\; = \;\; & 2\nabla_r g(Y(n);r)(f(Y(n)) - g(Y(n);r)) \\ & -2(f(Y(n)) - g(Y(n);r)) \textstyle\sum_{i=1}^{n} \lambda^{n-i} \nabla_r g(Y(i);r) \\ & -2\nabla_r g(Y(n);r) \textstyle\sum_{i=1}^{n} \lambda^{n-i} (f(Y(i)) - g(Y(i);r)), \qquad n \ge 1. \end{aligned} \tag{35}$$

Although it is not convergent, (35) is an asymptotically unbiased estimator of $\nabla s_\lambda(r)$ as $n \to \infty$. Therefore, (35) can be used as an updating term of a stochastic approximation algorithm searching for the minimum of $s_\lambda(\cdot)$: in that case the algorithm itself performs "averaging" of the gradient estimates (35), which is virtually equivalent to the strong convergence of an estimator. As a result of these facts, we get the following recursive algorithm for the simultaneous estimation of $\bar{f}$ and search for the zeros of $\nabla s_\lambda(\cdot)$:

Simulation Algorithm Using Adaptive Shadow Functions

For a given $\lambda \in [0,1)$, a given gain-sequence $\{\gamma_n : n \geq 1\}$ of positive numbers, and an initial value $r_0 \in \mathbb{R}^p$, estimates $\{\bar{f}_n\}$ of $\bar{f}$ are given for $n \geq 1$ by,

$$\bar{f}_n = n^{-1} \sum_{i=0}^{n-1} (f(Y(i+1)) - g(Y(i+1); r_i)). \tag{36}$$

The sequence $\{r_n\}$ is defined by the set of recursions, for $n \geq 0$,

$$\begin{aligned}
r_{n+1} &= r_n + \gamma_{n+1}(a_{n+1}u_{n+1} + b_{n+1}v_{n+1} - a_{n+1}b_{n+1}) && (37)\\
a_{n+1} &= \nabla_r g(Y(n+1); r_n) && (38)\\
b_{n+1} &= f(Y(n+1)) - g(Y(n+1); r_n) && (39)\\
u_{n+1} &= \lambda u_n + f(Y(n+1)) - g(Y(n+1); r_n) && (40)\\
v_{n+1} &= \lambda v_n + \nabla_r g(Y(n+1); r_n)\,, && (41)
\end{aligned}$$

where in (40), (41) we initialize $u_0 = 0$, $v_0 = \theta$.

The recursion defined by (37)–(41) falls into the category of stochastic gradient algorithms (which are a subclass of stochastic approximation algorithms; see e.g., [5, 38, 40, 53]).

The discount factor λ plays a stabilizing role in the recursions defining $\{u_n, v_n : n \geq 0\}$, and provides asymptotic "error forgetting" in the estimate of $\nabla s_\lambda(r_{n-1})$, given by

$$\widehat{\nabla} s_\lambda^n = 2(a_n b_n - a_n u_n - b_n v_n).$$

Under mild conditions, such as A1–A4 below, the sequence $\{r_n\}$ converges to a value $r_\lambda^* \in \mathbb{R}^p$ minimizing $s_\lambda(\cdot)$.

A1 The gain sequence satisfies,

$$\lim_{n\to\infty} \gamma_n = 0; \quad \sum_{n=1}^{\infty} \gamma_n = \infty; \quad \textit{and} \quad \sum_{n=1}^{\infty} \gamma_n^2 < \infty.$$

A2 The underlying Markov chain is V-uniformly ergodic, and $f^4 \in L_\infty^V$.

A3 The shadow functions are Lipschitz and locally bounded: Setting $\phi := V^{1/4}$, we assume that for some constant $L \in [1, \infty)$,

$$\begin{aligned}
|g(x;r)| &\leq L\phi(x)(1 + \|r\|),\\
\|\nabla_r g(x;r)\| &\leq L\phi(x),\\
\|\nabla_r g(x;r') - \nabla_r g(x;r'')\| &\leq L\phi(x)\|r' - r''\|
\end{aligned}$$

for all $r, r', r'' \in \mathbb{R}^p$, $x \in \mathsf{X}$.

A4 The "ODE at ∞" is stable: For any $r \in \mathbb{R}^p$ write,

$$S_\lambda(r) := \lim_{t\to\infty} t^{-1}\nabla s_\lambda(tr)$$

Assume that this is well-defined and finite for all $r \in \mathbb{R}^p$, and that $r = \theta$ is a globally asymptotically stable equilibrium for the ODE

$$\frac{d}{dt}\varrho(t) = -S_\lambda(\varrho(t)), \qquad \varrho(0) \in \mathbb{R}^p.$$

Assumption A1 is standard in the theory of stochastic approximation (see e.g., [5, 38, 40, 53]). For example, this condition holds for the specific choice $\gamma_n = n^{-\delta}$, $n \geq 1$, for any constant $\delta \in (1/2, 1]$.

Assumption A2 is related to the stability of the Markov chain $\boldsymbol{Y}$, while A3 corresponds with smoothness of the function g. These two assumptions are of crucial importance for both the derivation and analysis of the algorithm. In particular, they ensure that the estimator $\{\bar{f}(n; r)\}_{n\geq 1}$ is strongly consistent, and satisfies a central limit theorem for any fixed $r \in \mathbb{R}^p$.

Finally, Assumption A4 is introduced to establish pathwise boundedness of the algorithm when r is not fixed.

Analysis of the algorithm begins with the following decomposition for the gradient:

$$\widehat{\nabla} s_\lambda^{n+1} = \nabla s_\lambda(r(n)) + \xi_{n+1} + \xi'_{n+1} + (\xi''_{n+1} - \xi''_n), \qquad n \geq 0\,,$$

where $\{\xi_n\}_{n\geq 1}$ is a martingale difference sequence satisfying

$$\sup_{n\geq 0} \mathsf{E}[(1 + \|r(n)\|)^{-2}\|\xi_{n+1}\|^2] < \infty,$$

while $\{\xi'_n, \xi''_n : n \geq 0\}$ fulfill

$$\begin{aligned} \textstyle\sum_{n=0}^{\infty}(1 + \|r(n)\|)^{-1}\|\xi'_{n+1}\| &< \infty \quad \text{a.s.} \\ \sup_{n\geq 0} \mathsf{E}[(1 + \|r(n)\|)^{-2}\|\xi''_{n+1}\|^2] &< \infty. \end{aligned}$$

Consequently, a stability analysis can be carried out using the methods of [7]. Convergence of the algorithm then follows from well-worn arguments (see [5]).

For complete proofs of Theorems 3.1, 3.2 and further results see [59].

Theorem 3.1 *Suppose that A1-A4 hold. Then,*

(i) *$s_\lambda(\cdot)$ is C^1 and $\nabla s_\lambda(\cdot)$ is globally Lipschitz continuous.*

(ii) *The algorithm is stable, and convergent in the sense that*

$$\sup_n \|r_n\| < \infty, \quad \textit{and} \qquad \lim_{n\to\infty} d(r_n, E_\lambda^*) = 0, \qquad \textit{a.s.},$$

where $E_\lambda^* = \{r \in \mathbb{R}^p : \nabla s_\lambda(r) = 0\}$.

(iii) *Suppose that E^*_λ is a singleton, $E^*_\lambda = \{r^*_\lambda\}$. Then, the algorithm is convergent, and the normalized estimates*

$$n^{\frac{1}{2}}(\bar{f}_n - \bar{f}), \qquad n \geq 1,$$

converge in distribution to a Gaussian random variable with zero-mean, and variance $s_1(r^*_\lambda)$.

In many examples we can construct a linear parametrization, as discussed in Section 3.2. Suppose that $G : \mathsf{X} \to \mathbb{R}^p$ is a given Borel-measurable function satisfying $\|G\|^4 \in L^V_\infty$ and

$$\int G_i(x)\pi(dx) = 0, \qquad 1 \leq i \leq p,$$

where $G_i(\cdot)$ is the ith component of $G(\cdot)$. We may then set $g(x; r) = r^T G(x)$ for $r \in \mathbb{R}^p$, $x \in \mathsf{X}$, to obtain a function satisfying A3 and A4. Moreover, for any $\lambda \in [0, 1]$, $r \in \mathbb{R}^p$, the properties of the discounted TAVC $s_\lambda(r)$ can then be characterized in terms of the $p \times p$ matrix,

$$\Gamma_\lambda := \int G(x)G^T(x)\pi(dx) + \sum_{n=1}^{\infty} \lambda^n \int\int (G(x)G^T(y) + G(y)G^T(x))P^n(x, dy)\pi(dx).$$

Theorem 3.2 *Suppose that Assumptions A1 and A2 hold, and that $\|G\|^4 \in L^V_\infty$. Assume that Γ_1 is positive definite. (It is finite due to Assumptions A1, A2 and the fact that $\|G\|^4 \in L^V_\infty$.) Then,*

(i) *The discounted TAVC is finite, and the optimizer r^*_λ that minimizes $s_\lambda(\cdot)$ is unique for each $\lambda \in [0, 1]$. Moreover, there exists $M \in [1, \infty)$, independent of λ, such that*

$$\begin{aligned} \|r^*_1 - r^*_\lambda\| &\leq M(1 - \lambda), \\ 0 \leq s_1(r^*_\lambda) - s_1(r^*_1) &\leq M(1 - \lambda)^2 \end{aligned}$$

(ii) *The algorithm is consistent, in the sense that $r_n \to r^*_\lambda$ a.s. for any initial conditions, and the normalized estimates*

$$n^{\frac{1}{2}}(\bar{f}_n - \bar{f}), \qquad n \geq 1,$$

converge in distribution to a Gaussian random variable with zero-mean, and variance $s_1(r^*_\lambda)$.

3.5 Adaptive shadow functions for the 2-parameter queue

We provide here results for the G/G/1 queue modelled using the 2-parameter recursion (3). The application of adaptive shadow functions on more complex models is the subject of current research.

For a given μ, α the ratio $\rho = \alpha/\mu$ is equal to the load, which is assumed to be strictly less than unity. In all of our simulations of the single queue we have taken $\boldsymbol{B}, \boldsymbol{A}$

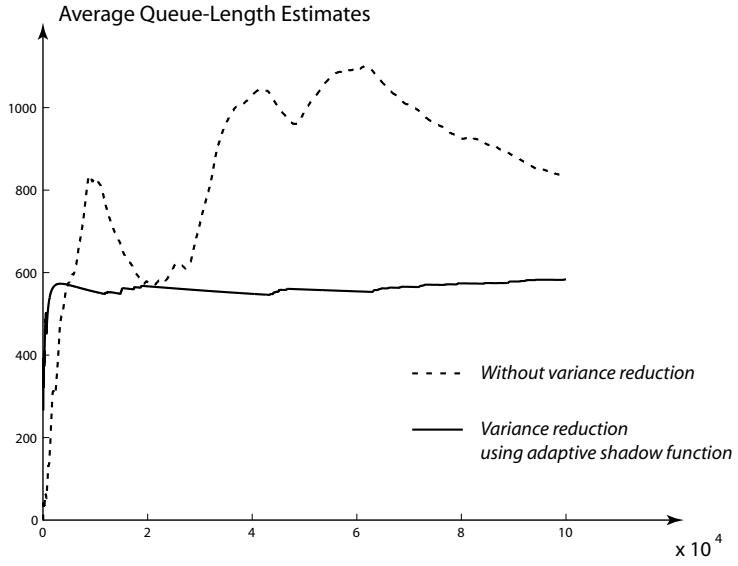

Figure 6: Results for a simulation run of length 100,000 steps for the single queue. The dashed line represents the running average cost using the standard estimator (24). The solid line represents the running average cost for the smoothed estimator considered in Theorem 3.2.

mutually independent and i.i.d., and the respective marginal distributions were taken to be a discrete distribution of the form (4): For any $k \geq 1$,

$$B(k) = \begin{cases} -\overline{\mu} & \text{with prob. } p_B = \mu/\overline{\mu}.; \\ 0 & \text{with prob. } 1 - p_B, \end{cases} \qquad A(k) = \begin{cases} \overline{\alpha} & \text{with prob. } p_A = \alpha/\overline{\alpha}.; \\ 0 & \text{with prob. } 1 - p_A, \end{cases}$$

$$\overline{\mu} = (1+\kappa)\mu \qquad \overline{\alpha} = (1+\kappa)\alpha$$

where $\sqrt{\kappa}$ is equal to the common coefficient of variation for these random variables.

The associated fluid model is given by (5), which in this simple case takes the form,

$$\frac{d}{dt} q(t;x) = -\mu\zeta(t) + \alpha, \qquad x \in \mathsf{X} = [0,\infty),\ t \geq 0.$$

Assuming that $\boldsymbol{\zeta}$ is defined by the non-idling policy, the fluid value function using $c(x) = x$ is given by

$$J(x) := \int_0^\infty c(q(t;x))\, dt = \tfrac{1}{2} \frac{x^2}{\mu - \alpha}.$$

We cannot compute the solution to Poisson's equation (22), but we do know that it satisfies,

$$\lim_{x\to\infty} \frac{h(x)}{J(x)} = 1.$$

Moreover, in many instances the error $h - J$ is a linear function of x (see e.g. (15)). This motivates a parameterized family of shadow functions of the form (32) with

$$h(x;r) = J(x) + rx.$$

Since all of our numerical results were consistent we present results from just one experiment using $\kappa = 10$, $\mu = 20/6$, $\alpha = 19/6$, so that the load is equal to $\rho = 95/100$. Figure 6 contains two plots: the first shows $\bar{f}(n)$ vs. n using the standard estimator (24), and the second plot shows a plot of $\bar{f}_n$ obtained using the estimator (36) with g of the form (32). Note that after 10^5 samples the standard estimator gives results that are nearly meaningless, while the refined estimator locks on to a steady state value of approximately 600.

While we do not know the actual steady state mean, we can compare these results with the steady-state mean for the associated RBM model with identical second order statistics. The steady-state distribution is exponential in this case, and the steady state mean is given by,

$$\mathsf{E}_\pi[Q(t)] = \tfrac{1}{2}\frac{\sigma_B^2 + \sigma_A^2}{\mu - \alpha} = \tfrac{1}{2}\frac{\kappa\mu^2 + \kappa\alpha^2}{\mu - \alpha} = 634\,\tfrac{2}{3}.$$

This is within approximately 5% of the value obtained in simulation for the 2-parameter model shown in Figure 6.

4 The Search for Safety-Stocks

In Section 2 we described approaches to policy synthesis through consideration of the fluid model. The affine policy (13) for the KSRS network is one example. To obtain an effective design one must select appropriate safety-stock levels: The best value will depend on network variability and load.

Shown in Figure 7 are estimates of the steady-state customer population for the KSRS model. The estimates were obtained as in Figure 5, with simulations run for a family of policies indexed by the safety-stock level $\bar{w} \in \mathbb{R}^2_+$. Figure 7 (i) shows results obtained using the smoothed estimator (26), and (ii) contains results obtained from the standard estimator (24). In these experiments we did not use the adaptive algorithm described in the previous section, instead opting for the static shadow function (25). Even without this additional refinement we see in Figure 7 (i) a remarkable improvement over the standard estimator.

Restricting attention to the results shown in Figure 7 (i), we first note that performance is approximately convex, with performance rapidly deteriorating for small values of either component of $\bar{w}$. Performance again deteriorates for larger values, but at a more gentle rate. The optimal value of 17.6 in this plot occurs at $\bar{w} = (35, 30)^T$. By way of contrast, the safety-stock value $\bar{w} = (60, 60)^T$ yields a cost of 18.8.

Recall that safety-stock values in (13) are measured in terms of *workload* at each resource, and not in terms of buffer levels. A "safety workload" of 35 at station 1 is perhaps difficult to interpret. In terms of buffer levels, this corresponds to approximately 11 jobs in buffer 1 (and 0 in buffer 4), or 4 jobs in buffer 4 (and 0 in buffer 1).

Given the symmetry of the model, one would expect the optimal point to lie on the diagonal $\{\bar{w}_1 = \bar{w}_2\}$. The asymmetry arises because the results from simulation are not exact. We could have exploited the symmetry of our chosen parameters for the KSRS model in constructing this plot, but opted not to do so since such symmetry will not be present in most models.

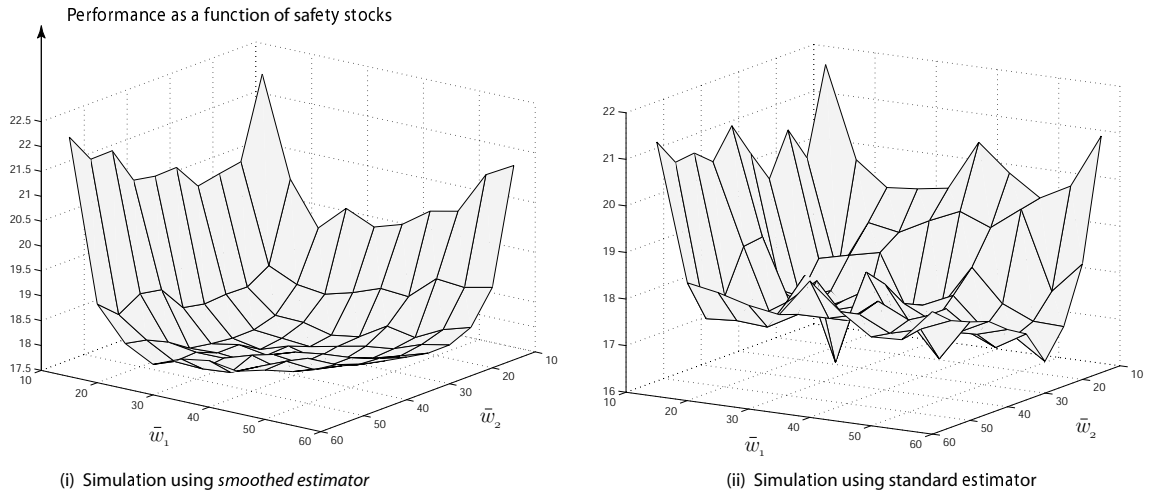

Figure 7: Estimates of the steady-state customer population in the KSRS model as a function of 100 different safety-stock levels. Two simulation experiments are shown, where in each case the simulation runlength consisted of 200,000 steps. The left hand side shows the results obtained using the smoothed estimator; the right hand side shows results with the standard estimator.

The convexity evident in Figure 7 suggests an efficient cutting plane method for searching for optimal safety-stock levels. This approach to convex optimization, originally proposed in [32], is based on approximaton of the convex function to be optimized by a succession of piecewise-linear convex functions. At each iteration of the algorthm, the piecewise-linear approximation is optimized using a linear program; the objective function and its gradient are evaluated at the minimizer; and the corresponding tangent plane is added as a cut to the linear program, thereby refining the piecewise linear approximation. This procedure is iterated until lower and upper bounds on the optimal objective function value are satisfactorily close. The upper bound is given by the value of the best solution seen thus far. The lower bound is given by the optimal value of the most recently solved linear program.

Let $\bar{w}_l$ and $\bar{w}_u$ be known (componentwise) upper and lower bounds on $\bar{w}$, so that $\bar{w}_l \leq \bar{w} \leq \bar{w}_u$. The cutting plane method successively adds cuts to the following base linear program, which includes a variable y that represents the piecewise linear approximation to the true convex function.

$$\begin{aligned} \textbf{min} \quad & y \\ \textbf{subject to} \quad & \bar{w}_l \leq \bar{w} \leq \bar{w}_u. \end{aligned} \tag{42}$$

The optimal solution to this linear program is $-\infty$, which reflects the fact that we have not yet added any cuts to restrict the value of y. The algorithm, specialized to our case of hunting for optimal safety-stock levels $\bar{w}$, is as follows.

1. Set $LB = -\infty, UB = +\infty$, $k = 0$, $\bar{w}_0 = \bar{w}_l$.
2. Compute an estimate of the performance z_k of the KSRS model with

safety-stock levels $\bar{w}_k$, along with an estimate of the gradient g_k of performance with respect to the safety-stock levels. Set $UB = \min(UB, z_k)$.

3. Add the cut to the linear program:

$$y \geq z_k + g_k^T(\bar{w} - \bar{w}_k)$$

4. Set $k = k + 1$.
5. Solve the new linear program. Let $(\bar{w}_k, y_k)$ be the new optimal solution. Set $LB = y_k$.
6. If $UB - LB$ is larger than a user specified tolerance then go to Step 2.

This algorithm relies on estimates of the gradient of performance with respect to the safety-stocks. This suggests the application of single-run gradient estimation methods, such as infinitesimal perturbation analysis. Unfortunately, it is shown in [28] that this approach, or its extensions, cannot be efficiently implemented in our setting. So instead we rely on finite-difference approximations of the gradient.

The performance of the cutting plane algorithm is depicted in Table 4. In order to reuse the extensive simulation results depicted in Figure 7, we modified the above algorithm slightly. After obtaining $\bar{w}_k$ in Step 5 of the algorithm, we rounded each of the components to the nearest multiple of 5. Furthermore, we obtained the gradient estimates by using forward differences of size 5, so as to align the calculations with the grid of values given in Figure 7. Initial simulations suggested taking $\bar{w}_l = (15, 15)^T$ and $\bar{w}_u = (55, 55)^T$. The final solution reported by this algorithm, $\bar{w}_8 = (35, 30)^T$, is indeed the minimal value in Figure 7.

It is perhaps surprising that the lower bound exceeds the upper bound in Iteration 8. This arises because the convexity assumption is violated slightly in estimates of the function values.

k	$\bar{w}_k$	LB	UB
0	$(15, 15)$	$-\infty$	22.14
1	$(55, 55)$	-6.6	18.46
2	$(35, 15)$	16.35	18.46
3	$(20, 25)$	16.43	18.31
4	$(35, 25)$	16.76	17.72
5	$(30, 40)$	17.31	17.72
6	$(40, 40)$	17.51	17.72
7	$(30, 30)$	17.55	17.68
8	$(35, 30)$	17.64	17.62

Table 1: Performance of the cutting plane algorithm on the KSRS example.

The cutting plane algorithm required 27 function evaluations (3 per iteration) to obtain this solution as compared with the 100 function evaluations used to construct Figure 7. The savings in computation could be even greater if one were to store function evaluations in a table and only run simulations if need be. Note also that great gains are made in the initial steps of the algorithm, but these gains diminish quickly after a few iterations. This

behavior is often seen in cutting plane algorithm applications, and suggests that great efficiencies are possible if one is willing to accept approximate solutions.

This example, while admittedly contrived, serves to illustrate the potential of the cutting plane method in conjunction with simulation. The optimization problem is somewhat "easy", being very low dimensional. In higher dimensional problems involving more resources and hence more safety-stock level variables, one would expect even greater savings with the cutting plane method over brute-force experimentation.

The cutting plane method relies heavily on convexity, or near convexity, of the underlying performance measure. The examples in [12] and [28] support the notion that performance as a function of safety-stocks is convex, or quasi-convex, at least near the optimal choice of safety-stocks.

We have omitted any discussion of the fact that we are using simulation-based estimates of function values and gradients, as opposed to actual values. For an in-depth discussion of this point and its consequences, see [2].

5 Conclusions and Future Research

Network control remains a central problem in a myriad of applications. The large scale and complexity of these networks ensures that optimal control is, practically speaking, beyond reach. However, we can hope to develop control algorithms that are flexible, easily implemented and, in many cases, "nearly" optimal.

We have developed the concept of a 2-parameter model. Like Brownian models, they allow independent modelling of variability and drift. Unlike Brownian models, solutions always exist, and performance evaluation is straightforward through simulation.

The concept of a workload relaxation allows us to develop optimal, or near-optimal, policies for fluid models that can then serve as a starting point for developing policies in the original discrete setting. A key issue in this translation is the development of safety-stocks. We have shown how to use cutting plane methods, in conjunction with simulation, to estimate optimal safety-stock levels.

Workload relaxations also provide approximations to the fluid value function that can then be used to obtain tremendous variance reductions in simulations of network behavior. These variance reductions are essential if one is to use simulation to search for appropriate safety-stock levels. The value function approximations can be "tuned" using a stochastic approximation algorithm.

Many open issues remain:

- Is it possible to perform on-line optimization in more realistic models? Can parameters (e.g., safety-stock levels) be tuned in "real-time"?
- To what extent can these methods be extended to non-standard performance metrics such as might arise in capturing, for example, economic issues of price/demand fluctuations, and robustness to resource outages?
- We have assumed that information is centralized. How might these ideas be extended to situations in which information is distributed?

- We have only considered safety-stocks in translating from the fluid to the discrete domain. Several other issues remain, such as the need to avoid excessive setups in switching resources between tasks. Are there general methods for performing this translation in an effective manner? Can efficient optimization techniques (like cutting plane methods) be employed in this setting?

- Can the powerful combination of stochastic approximation and variance reduction be exploited to an even greater extent than was done here?

We are currently considering all of these issues, as well as application to large-scale manufacturing; communication systems; and resource allocation in electrical power markets.

References

[1] S. Asmussen. Queueing simulation in heavy traffic. *Math. Operations Res.*, 17:84–111, 1992.

[2] J. Atlason, M. A. Epelman, and S. G. Henderson. Combining simulation and cutting plane methods in service systems. In P. Reinig, editor, *Proceedings of the 2002 National Science Foundation Design, Service and Manufacture Grantees and Research Conference*, 2002.

[3] N. Baëuerle. Asymptotic optimality of tracking-policies in stochastic networks. Annals of Applied Probability, To appear., 1999.

[4] S.L. Bell and R.J. Williams. Dynamic scheduling of a system with two parallel servers: Asymptotic optimality of a continuous review theshold policy in heavy traffic. In *Proceedings of the 38th Conference on Decision and Control*, pages 1743–1748, Pheonix, Arizona, 1999.

[5] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*. Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.

[6] D. Bertsimas, I. Paschalidis, and J. N. Tsitsiklis. Optimization of multiclass queueing networks: polyhedral and nonlinear characterizations of achievable performance. *Ann. Appl. Probab.*, 4:43–75, 1994.

[7] V. S. Borkar and S. P. Meyn. The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.*, 38(2):447–469 (electronic), 2000.

[8] M. Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems*, 30:89–148, 1998.

[9] M. Bramson and J. G. Dai. Heavy traffic limits for some queueing networks. *Ann. Appl. Probab.*, 11(1):49–90, 2001.

[10] H. Chen and A. Mandelbaum. Stochastic discrete flow networks: diffusion approximations and bottlenecks. *Ann. Probab.*, 19(4):1463–1519, 1991.

[11] H. Chen and D. D. Yao. Dynamic scheduling of a multiclass fluid network. *Operations Research*, 41(6):1104–1115, November-December 1993.

[12] M. Chen, C. Pandit, and S. P. Meyn. In search of sensitivity in network optimization. Submitted for publication, and presented at the 11TH INFORMS Applied Probability Society Conference, 2001.

[13] R-R. Chen and S. P. Meyn. Value iteration and optimization of multiclass queueing networks. *Queueing Systems Theory Appl.*, 32(1-3):65–97, 1999.

[14] P. J. Courtois. *Decomposability*. Academic Press, New York, 1977.

[15] J. G. Dai and Y. Wang. Nonexistence of Brownian models of certain multiclass queueing networks. *Queueing Systems: Theory and Applications*, 13:41–46, May 1993.

[16] A. Dembo and O. Zeitouni. *Large deviations techniques and applications*. Springer-Verlag, New York, second edition, 1998.

[17] R. Dubrawski. Myopic and far-sighted strategies for control of demand-driven networks. Master's thesis, Department of Electrical Engineering, UIUC, Urbana, Illinois, USA, 2000.

[18] P. Dupuis and H. Kushner. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer Verlag, New York, 2001. Applications of Mathematics, Vol 24.

[19] E. A. Feinberg and A. Shwartz. *Handbook of Markov Decision Processes*. Kluwer Academic Publishers, Boston, 2001.

[20] P. W. Glynn and W. Whitt. Indirect estimation via $l = \lambda w$. *Operations Research*, 37(1):82–103, 1989.

[21] J. M. Harrison and J. A. Van Mieghem. Dynamic control of Brownian networks: state space collapse and equivalent workload formulations. *Ann. Appl. Probab.*, 7(3):747–771, 1997.

[22] J. M. Harrison and L. M. Wein. Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Operations Research*, 38(6):1052–1064, 1990.

[23] J.M. Harrison. *The BIGSTEP approach to flow management in stochastic processing networks*, pages 57–89. Stochastic Networks Theory and Applications. Clarendon Press, Oxford, UK, 1996. F.P. Kelly, S. Zachary, and I. Ziedins (ed.).

[24] S. G. Henderson and P. W. Glynn. Approximating martingales for variance reduction in general discrete-event simulation. *Working paper*, 2001.

[25] S. G. Henderson and P. W. Glynn. Approximating martingales for variance reduction in Markov process simulation. *Mathematics of Operations Research*, 2002. To appear.

[26] S. G. Henderson and S. P. Meyn. Efficient simulation of multiclass queueing networks. In S. Andradottir, K. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 216–223, Piscataway, N.J., 1997. IEEE.

[27] S. G. Henderson and S. P. Meyn. Variance reduction for simulation in multiclass queueing networks. Submitted for publication, 1999.

[28] S. G. Henderson and S. P. Meyn. Identifying effective polices for multiclass networks. In P. Reinig, editor, *Proceedings of the 2002 National Science Foundation Design, Service and Manufacture Grantees and Research Conference*, 2002.

[29] S.G. Henderson. *Variance Reduction Via an Approximating Markov Process*. PhD thesis, Department of Operations Research, Stanford University, Stanford, California, USA, 1997.

[30] A. Hordijk and F. Spieksma. On ergodicity and recurrence properties of a Markov chain with an application to an open Jackson network. *Adv. in Appl. Probab.*, 24(2):343–376, 1992.

[31] N. V. Kartashov. Criteria for uniform ergodicity and strong stability of Markov chains with a common phase space. *Teor. Veroyatnost. i Mat. Statist.*, pages 65–81, 151, 1984.

[32] J.E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[33] I. Kontoyiannis and S.P. Meyn. Spectral theory and limit theorems for geometrically ergodic Markov processes. *Ann. Appl. Probab.*, 2002. Presented at the 2001 INFORMS Applied Probability Conference, NY, July, 2001.

[34] P. R. Kumar and S. P. Meyn. Duality and linear programs for stability and performance analysis queueing networks and scheduling policies. *IEEE Transactions on Automatic Control*, 41(1):4–17, 1996.

[35] S. Kumar and P. R. Kumar. Performance bounds for queueing networks and scheduling policies. *IEEE Trans. Automat. Control*, AC-39:1600–1611, August 1994.

[36] S. Kumar and M. Muthuraman. A numerical method for solving singular Brownian control problems. In *Proceedings of the 39th Conference on Decision and Control*, 2000.

[37] H. J. Kushner. *Heavy traffic analysis of controlled queueing and communication networks*. Springer-Verlag, New York, 2001. Stochastic Modelling and Applied Probability.

[38] H. J. Kushner and G. George Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, New York, 1997.

[39] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 3rd edition, 2000.

[40] L. Ljung, G. Pflug, and H. Walk. *Stochastic approximation and optimization of random systems*. Birkhäuser Verlag, Basel, 1992.

[41] X. Luo and D. Bertsimas. A new algorithm for state-constrained separated continuous linear programs. *SIAM J. Control Optim.*, 37:177–210, 1998.

[42] C. Maglaras. Design of dynamic control policies for stochastic processing networks via fluid models. In *Proceedings of the 38th Conference on Decision and Control*, pages 1208–1213, 1997.

[43] C. Maglaras. Dynamic scheduling in multiclass queueing networks: Stability under discrete-review policies. *Queueing Systems*, 31:171–206, 1999.

[44] C. Maglaras. Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality. *Ann. Appl. Probab.*, 10:–, 2000.

[45] S. P. Meyn. Columbia University lecture series on stochastic networks. Center for Applied Probability, Columbia University, New York, NY, July, 2001. http://black.csl.uiuc.edu/~meyn/pages/ColumbiaLectures.pdf.

[46] S. P. Meyn. The policy iteration algorithm for average reward Markov decision processes with general state space. *IEEE Trans. Automat. Control*, 42(12):1663–1680, 1997. Also presented at the *35th IEEE Conference on Decision and Control*, Kobe, Japan, Dec., 1996.

[47] S. P. Meyn. Stability and optimization of queueing networks and their fluid models. In *Mathematics of Stochastic Manufacturing Systems (Williamsburg, VA, 1996)*, pages 175–199. Amer. Math. Soc., Providence, RI, 1997.

[48] S. P. Meyn. Sequencing and routing in multiclass queueing networks. Part I: Feedback regulation. *SIAM J. Control Optim.*, 40(3):741–776, 2001.

[49] S. P. Meyn. Sequencing and routing in multiclass queueing networks. Part II: Workload relaxations. *SIAM J. Control Optim.*, 2002. To appear.

[50] S. P. Meyn and D. G. Down. Stability of generalized Jackson networks. *Ann. Appl. Probab.*, 4:124–148, 1994.

[51] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993. Available on-line at http://black.csl.uiuc.edu/~meyn.

[52] J. R. Morrison and P. R. Kumar. New linear program performance bounds for queueing networks. *Journal of Optimization Theory and Applications*, 100(3):575–597, March 1999. Plenum Publishers.

[53] M. B. Nevel′son and R. Z. Has′minskiĭ. *Stochastic approximation and recursive estimation*. American Mathematical Society, Providence, R. I., 1973. Translated from the Russian by the Israel Program for Scientific Translations, Translations of Mathematical Monographs, Vol. 47.

[54] J. Perkins. *Control of Push and Pull Manufacturing Systems*. PhD thesis, University of Illinois, Urbana, IL, September 1993. Technical report no. UILU-ENG-93-2237 (DC-155).

[55] M. I. Reiman. Open queueing networks in heavy traffic. *Mathematics of Operations Research*, 9:441–458, 1984.

[56] M. I. Reiman. Some diffusion approximations with state space collapse. In *Modelling and performance evaluation methodology (Paris, 1983)*, pages 209–240. Springer, Berlin, 1984.

[57] S. M. Ross. *Stochastic Processes*. Wiley, New York, 2nd edition, 1996.

[58] E. Schwerer. A linear programming approach to the steady-state analysis of reflected Brownian motion. *Stochastic Models*, 17:341–368, 2001.

[59] V. Tadic and S.P. Meyn. A stochastic approximation algorithm for variance reduction in simulation and reinforcement learning. In preparation, 2002.

[60] G. Weiss. A simplex based algorithm to solve separated continuous linear programs. Technical Report,Department of Statistics The University of Haifa Mount Carmel 31905, Israel, 2001.

[61] W. Whitt. Planning queueing simulations. *Management Science*, 35:1341–1366, 1994.

[62] R. J. Williams. Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse. *Queueing Systems*, 30(1-2):27–88, 1998.