Applied Stochastic
Models in Business
and Industry

Research Article

Received 13 February 2012, Revised 12 November 2012, Accepted 26 November 2012 Published online 27 December 2012 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/asmb.1961

Estimation and monitoring of traffic intensities with application to control of stochastic systems

Ying-Chao Hung[a*†], **George Michailidis**[b] **and Shih-Chung Chuang**[c]

The development of optimal control strategies for many stochastic models relies on the observed traffic intensity. However, implementation of such control strategies is often infeasible because of high operating costs induced by the fluctuations of traffic flows. In this study, we propose a framework for estimating and monitoring the traffic intensities of stochastic systems. The framework does not require knowledge of any input traffic statistics, and it allows us to adaptively estimate the intensity function over time and simultaneously detect its significant changes so that the control strategy can be adjusted accordingly without requiring high operating costs. Finally, a canonical queueing system with various types of input traffic is used to evaluate the effectiveness of the proposed framework. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords: traffic intensity; EWMA smoother; control chart; control policy; stochastic systems

1. Introduction and a motivating application

This research is motivated by a control problem for the following canonical queueing model. There are Q classes of customers (jobs) stored in dedicated first-in-first-out queues and a server with parallel processing capabilities. Customers (jobs) arrive at each queue according to a stochastic process of rate $\lambda_q,\ q = 1, \ldots, Q$. The queues have infinite capacity buffers, where jobs are placed while waiting to receive service. The service mechanism is also stochastic, say, suppose there are M service modes, with mode m associated with a Q-dimensional vector $R_m = (\mu_{m1}, \cdots, \mu_{mQ})$ that determine at which rate jobs of different classes are processed. At any point in time t, each service mode m can be possibly employed with a level $u_m(t)$, $0 \leqslant u_m(t) \leqslant 1$ for all $m = 1, \ldots, M$, and $\sum_{m=1}^{M} u_m(t) \leqslant 1$. Such a service scheme has an important feature that all service resources (or capabilities) can be shared (in various proportions) simultaneously, which was first introduced by Hung and Michailidis [1]. A schematic representation of the system under consideration is shown in Figure 1.

This canonical model captures the fundamentals of many real stochastic systems such as call centers, cluster computing, wireless networking, flexible manufacturing, and so on [2, 3]. For example, in call centers, the different customer classes correspond to different types of service requests (e.g., reservations for flights, hotels, cars, and vacation packages), and the service modes are determined by the levels of specialization of the workforce; that is, there is a fixed set of agents with different skills and expertise that depending on which customer class they are assigned to, they determine the corresponding service rate. In cluster computing, the customer classes correspond to different types of program submitted to the system, and the server is comprised by a set of available resources (CPUs, storage units, memory modules, graphics cards, etc.). In wireless networking, the customers are requests (phone calls or data transmissions) on different channels, and the server corresponds to a base station transmitting to the customers at different power levels. In flexible manufacturing, the classes correspond to different products and the server to a set of reconfigurable tools building (serving) those products; the service modes are determined by different tool configurations that in turn specify the service rates.

The fundamental issue in such a complex queueing system is to decide how to allocate/control the service capabilities to maximize/minimize some performance metrics of interest, such as throughput, the average delay over all customer classes,

[a] *Department of Statistics, National Chengchi University, Taipei 11605, Taiwan*
[b] *Department of Statistics and Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-1092, USA*
[c] *Department of Industrial Engineering and Engineering Management, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Rd., Hsinchu 30013, Taiwan*
**Correspondence to: Ying-Chao Hung, Department of Statistics, National Chengchi University, Taipei 11605, Taiwan.*
[†] *E-mail: hungy@nccu.edu.tw*

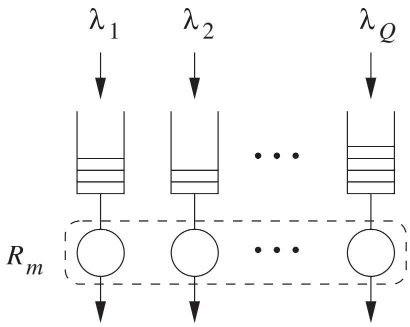

Figure 1. A schematic representation of the queueing system under consideration.

or the average backlog (jobs waiting in the queues). Such a service-allocation strategy, if exists, is called an optimal control policy. For the system under consideration, the control policy corresponds to best determining the level $u_m(t)$ for each service mode R_m at any point in time t. It should be noted that such policies are easy to construct under the assumption of regulated traffic flows and a suitable formulation of fluid models with respect to some QoS metrics (see [1, 4, 5] and the example in Section 3). Therefore, they can act as alternatives to the rather complicated policies presented in literature [2, 3, 6–16]. An important feature of such control policies is that they depend crucially on the *online estimation* of the current input vector. This suggests that for systems with time-varying traffic intensities, the policy has to be more responsive to traffic fluctuations to achieve better performance. Hence, estimation and prediction of instantaneous traffic intensities become necessary. Recently, Aktekin and Soyer [17] and Weinberg *et al.* [18] utilized Bayesian approaches to estimate and predict the inter-day and intra-day arrival rates of a call center. The shortcoming for this type of approaches is that small local shifts in the resulting estimation of intensities would also require a recalculation of the corresponding optimal control, which in turn would incur a certain computational/operational cost. To overcome these problems, we design a general framework that simultaneously provides the estimation (or one-step-ahead prediction) for the traffic intensity and monitors its 'significant' shifts over time. The framework basically extends the ideas from our previous work [1], which employs a simple smoothing technique called the exponentially weighted moving average (EWMA), and provides a practical guideline for choosing all relative tuning parameters. On the basis of the framework, a sequence of EWMA estimators for the intensity can be easily constructed by observing the traffic traces over time, whereas the optimal control is resolved only if 'significant' shifts in the estimated traffic intensity are detected. It should be noted that the proposed framework is purely measurement based and does not require knowledge of any input traffic statistics. Therefore, it is applicable to any types of input traffic (e.g., nonstationary input traffic).

Comprehensive statistical procedures have been extensively developed for the output analysis of queueing systems. For example, Asmussen [19] introduced an effective Monte Carlo approach involving importance sampling as well as linear regression for studying the mean and other functionals of queues; Hung *et al.* [20] developed a framework of efficient simulations for complex queueing systems by utilizing the idea of treed models and optimum design; Wieland *et al.* [21] proposed a simulation-based procedure for checking the stability of queues by utilizing the idea of statistical hypothesis testing; Kleijnen [22] introduced a regression model for analyzing the input–output behavior of the underlying complicated simulation model; and Cheng and Kleijnen [23] utilized the idea of optimum design to determine the traffic intensities at which the queueing systems are simulated so as to improve the proposed regression model for estimating the mean response of interest, just to name a few. The study of measurement-based optimal resource allocation problems for queueing systems is rather limited, for which some can be found in the recent works by Hayel *et al.* [24], Hung and Michailidis [25], Kallitsis *et al.* [26], and Xu *et al.* [27]. It should be highlighted that the work in this study, to the best of our knowledge, is the first attempt to integrate the statistical online estimation and monitoring strategies into the control formulation. In addition to queueing systems, it can be extensively applied to any stochastic models with similar flow analogies and control schemes, such as material flow control in the design of production lines [28], control of flow cytometry in cell biology [29–31], control of semi-active suspensions for vehicles or shock and vibration isolation systems [32], just to name a few.

The remainder of this paper is organized as follows. In Section 2, the framework based on the EWMA smoothing technique and the control chart used for monitoring the shifts of the estimated traffic intensity are introduced. In Section 3, a motivated fluid control problem for the queueing system under consideration is introduced. Further, the proposed framework is illustrated and its performance evaluated on a number of systems with various types of input traffic flows via a simulation study. Some concluding remarks are drawn in Section 4.

2. Framework for estimation and shift monitoring of traffic intensity

Suppose that the optimal control policy of the stochastic system requires knowledge of traffic intensities. This implies that for systems with time-varying traffic intensities, the control policy has to be more responsive to traffic fluctuations to achieve better performance. Hence, estimation (or prediction) and shift monitoring of instantaneous traffic intensities becomes necessary.

2.1. *Estimation of traffic intensity*

We estimate the traffic intensities over time by tracking the amount of work coming into the system. Because we do not place any structural assumptions on the input processes, a simple and popular technique for estimating the traffic intensity is the EWMA estimator. We next briefly describe the basics of the EWMA smoothing technique and how to adequately choose the tuning parameter.

Let us first focus on the traffic trace of a particular input flow and assume that it is a general marked point process. Suppose that time is divided into non-overlapping and equal-length intervals $((k-1)T, kT]$, $k = 1, 2, \ldots$, so that there are $N(k)$ arrivals during the kth time interval, with the jth job bringing a δ_j service requirement, $j = 1, \ldots, N(k)$. The first step is to calculate the 'mean' intensity of the traffic trace by aggregating all job arrivals over the kth time interval:

$$\bar{\lambda}(k) = \frac{1}{T}\sum_{j=1}^{N(k)} \delta_j, \quad k = 1, 2, \ldots. \tag{1}$$

Hence, a sequence of values $\left\{\bar{\lambda}(1), \bar{\lambda}(2), \ldots\right\}$ can be obtained. By treating $\bar{\lambda}(1), \bar{\lambda}(2), \ldots$ as an observed time series, the EWMA estimator is given by

$$\hat{\lambda}(k) = \beta\bar{\lambda}(k) + (1-\beta)\hat{\lambda}(k-1), \tag{2}$$

where $0 < \beta \leqslant 1$ gives the weight (or importance) used for the most recent observed traffic intensity. It is easy to show that $\hat{\lambda}(k)$ is the weighted average of all past k observed values, say,

$$\hat{\lambda}(k) = \beta\bar{\lambda}(k) + \beta(1-\beta)\bar{\lambda}(k-1) + \cdots + \beta(1-\beta)^{k-1}\bar{\lambda}(1). \tag{3}$$

Therefore, a normalized version of (3) can be expressed as

$$\hat{\lambda}(k) = \frac{\sum_{j=0}^{k-1}(1-\beta)^j\bar{\lambda}(k-j)}{\sum_{j=0}^{k-1}(1-\beta)^j}. \tag{4}$$

Adaptive choice of β

As it can be seen from (2), the EWMA estimator has a very simple algebraic formula. In many applications, it is extensively used as a one-step-ahead predictor of the traffic intensity. A popular choice for the optimal value of β is the one that minimizes the one-step-ahead mean squared prediction errors (MSPE). However, this requires knowledge of the underlying stochastic process or performing an intricate procedure for model selection. For example, Cox [33] obtained the optimal value of β for the AR(1) model; Montgomery and Mastrangelo [34] obtained the optimal value of β for general correlated data via simulation; and Ramjee *et al.* [35] obtained the optimal value of β for the FARIMA process. When the underlying process is unknown and statistics of the traffic trace can change over time, it is not clear how one can choose the best value of β so as to minimize the MSPE. Moreover, when the characteristics of the traffic flow can change over time, choosing a fixed value of β for the EWMA estimator might not be flexible enough to optimize the prediction accuracy.

Our goal here is to provide an adaptive choice for the value of β so that the EWMA estimator can predict well the fluctuations of any traffic trace. Recall the work performed by Montgomery and Mastrangelo [34]; the optimal value of β that minimizes the MSPE for an AR(1) model is given by $1-\frac{1}{2}(1-\phi)/\phi$ when the model parameter $\frac{1}{3} \leqslant \phi \leqslant 1$. As can be seen, the optimal value of β increases with the parameter ϕ. Further, a quick examination shows that $1-\frac{1}{2}(1-\phi)/\phi$ can be fairly well approximated by ϕ when $\frac{1}{2} \leqslant \phi \leqslant 1$. This means that when the underlying process is AR(1), the EWMA estimator may potentially perform well (in terms of prediction) by choosing $\beta = \phi$. Because we also know that for an

AR(1) model ϕ is equivalent to the lag one autocorrelation function, the optimal value of β can then be replaced by its natural estimate, say, suppose there are k observations $\bar{\lambda}(1), \ldots, \bar{\lambda}(k)$,

$$\hat{\rho}_1(k) = \frac{1}{(k-1)s^2} \sum_{i=1}^{k-1} \left[\bar{\lambda}(i+1) - \bar{\bar{\lambda}}(k)\right]\left[\bar{\lambda}(i) - \bar{\bar{\lambda}}(k)\right], \quad k \geqslant 2, \tag{5}$$

where $\bar{\bar{\lambda}}(k)$ and s are the mean and standard deviation of observations $\bar{\lambda}(1), \ldots, \bar{\lambda}(k)$, respectively.

From (2) and (3), it can be seen that the EWMA estimator is mainly contributed by the current observation and the previous one. Therefore, the idea of choosing the lag one autocorrelation function as the smoothing parameter seems quite intuitive. We next evaluate the performance of the EWMA estimator by choosing $\beta = \hat{\rho}_1(k)$ at each time kT for the following traffic traces: (i) the AR(1) model with different choices of model parameter ϕ; (ii) the MA(1) model with different choices of model parameter θ; (iii) the ARMA(2,1) model with different choices of model parameters ϕ and θ; (iv) the normalized fractional Brownian motion (FBM) with different choices of Hurst parameter H; and (v) a (normalized) real traffic trace collected from the Indianapolis and Cleveland links of the Abilene backbone network (see http://moat.nlanr.net/Images/Nvabar/2_MNA.html and [3, 36] for a complete description of this data set). The resulting MSPE along with those obtained by choosing different (but fixed) values of β are shown in Table I, where the minimal MSPE is highlighted in boldface.

As can be seen from Table I, the EWMA estimator with $\beta = \hat{\rho}_1(k)$ almost outperforms all other fixed choices of β for various types of input traffic (including short-range and long-range dependent processes). In some cases where $\beta = \hat{\rho}_1(k)$ does not perform best (compared to the minimal MSPE by choosing a fixed value of β), the numerical result reveals that it is still very competitive. Because $\hat{\rho}_1(k)$ is estimated by tracking the amount of work coming into the system, using $\beta = \hat{\rho}_1(k)$ instead of a fixed value is apparently more adaptive.

Remark 1

The EWMA estimator with $\beta = \hat{\rho}_1(k)$ is easy to construct and requires a rather small amount of computation. In addition, it can be applied to any types of input traffic, such as short-range/long-range dependent, seasonal/cyclical, and other nonstationary processes. In some special cases where $\hat{\rho}_1(k) \leqslant 0$, we can choose $\beta = \epsilon$, where ϵ is a positive value close to zero. The idea comes from the fact that when the observed value $\bar{\lambda}(k)$ is negatively correlated with the previous one $\bar{\lambda}(k-1)$, we prefer using a more conservative one-step-ahead predictor $\hat{\lambda}(k-1)$ (which is the weighted average of all previously observed $\bar{\lambda}(1), \ldots, \bar{\lambda}(k-1)$).

2.2. *Shift monitoring of traffic intensity*

Note that in practice, we can solve the optimal control on the basis of the estimated traffic intensity $\hat{\lambda}(k)$ at a sequence of selected points in time $\{T_1, T_2, T_3, \ldots\}$. However, if the duration between two selected time points (i.e. $T_{k+1} - T_k$) is long, the estimated intensity function will not be able to quickly respond to traffic fluctuations, thus inducing poor performance with respect to QoS metrics (such as delay). On the other hand, if the duration between two selected time points is short, the implementation of control policy becomes very sensitive, thus inducing high operating costs (e.g., a huge amount of recalculation for the optimal control or switching between control mechanisms). This implies that to best implement the control policy, one must take into account the trade-off between the performance measure and the computational/operational cost.

As we discussed, small local shifts in the resulting $\hat{\lambda}(k)$ would require a recalculation of the optimal control, which in turn would incur a certain computational/operational cost. To overcome this problem, we adopt a monitoring scheme for the estimated traffic intensities based on the EWMA control chart scheme. As indicated by the EWMA control chart (see later for details), the optimal control changes only if there is a 'significant' shift in the underlying traffic intensities. We next briefly summarize the monitoring strategy on the basis of such control charts.

Let $\left\{\bar{\lambda}(1), \bar{\lambda}(2), \ldots\right\}$ be a sequence of estimated traffic intensities (that could possibly be dependent) for a particular input flow. For practical purposes, here we consider a relaxed assumption that $\left\{\bar{\lambda}(1), \bar{\lambda}(2), \ldots\right\}$ are *wide sense stationary* with the true mean λ and covariance function $\gamma_j = \mathrm{Cov}\left(\bar{\lambda}(k), \bar{\lambda}(k-j)\right)$, $j = 0, \ldots, k-1$. Define vectors

$$A(k) = \frac{1}{\sum_{j=0}^{k-1} \prod_{i=0}^{j-1} [1-\beta(k-i)]\beta(k-j)} \begin{bmatrix} \beta(k) \\ [1-\beta(k)]\beta(k-1) \\ \prod_{i=0}^{1} [1-\beta(k-i)]\beta(k-2) \\ \vdots \\ \prod_{i=0}^{k-2} [1-\beta(k-i)]\beta(1) \end{bmatrix} \tag{6}$$

Table I. The estimated mean squared prediction errors (MSPE) based on the exponentially weighted moving average estimator with different choices of β for various traffic traces. The minimal MSPE obtained by choosing the best (but fixed) value of β is highlighted in boldface. Note that for each input source, the MSPE was calculated on the basis of 10^5 observations.

Source	Parameters	$\hat{\beta} = \hat{\rho}_1(k)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AR(1)	$\phi = 0.2$	0.3945	**0.3839**	0.3954	0.4089	0.4246	0.4427	0.4640	0.4889	0.5185	0.5539
	$\phi = 0.5$	0.4333	0.4684	0.4530	0.4425	0.4363	**0.4341**	0.4358	0.4415	0.4516	0.4665
	$\phi = 0.8$	0.4016	0.7557	0.6210	0.5387	0.4848	0.4485	0.4243	0.4090	0.4009	**0.3992**
MA(1)	$\theta = 0.2$	1.1122	**1.0791**	1.1168	1.1591	1.2065	1.2602	1.3215	1.3919	1.4741	1.5710
	$\theta = 0.5$	1.3072	**1.2661**	1.2808	1.2968	1.3146	1.3350	1.3584	1.3856	1.4174	1.4552
	$\theta = 0.8$	1.6407	**1.6383**	1.6394	1.6415	1.6442	1.6476	1.6516	1.6564	1.6619	1.6686
	$\theta = 1.0$	1.9832	1.9979	1.9946	1.9929	1.9919	1.9912	1.9906	1.9899	1.9891	**1.9882**
ARMA(2,1)	$\phi = (0.6, 0.3),\ \ \theta - 0.2$	1.0355	2.7393	1.9000	1.5154	1.3030	1.1756	1.0981	1.0540	1.0351	**1.0378**
	$\phi = (0.6, 0.3),\ \ \theta - 0.5$	1.0980	4.0889	2.7328	2.0941	1.7278	1.4945	1.3365	1.2259	1.1480	**1.0943**
	$\phi = (0.6, 0.3),\ \ \theta - 0.8$	1.3592	5.9069	3.9066	2.9556	2.4022	2.0427	1.7924	1.6099	1.4725	**1.3669**
	$\phi = (0.3, 0.1),\ \ \theta - 0.2$	1.1864	1.2748	1.2343	1.2072	1.1922	**1.1881**	1.1946	1.2121	1.2414	1.2843
	$\phi = (0.3, 0.1),\ \ \theta - 0.5$	1.3526	1.6717	1.5818	1.5080	1.4475	1.3985	1.3598	1.3304	1.3099	**1.2983**
	$\phi = (0.3, 0.3),\ \ \theta - 0.8$	1.6960	2.3136	2.1625	2.0344	1.9248	1.8306	1.7495	1.6797	1.6197	**1.5685**
FBM	H = 0.1	0.0208	0.0393	0.0295	0.0254	0.0232	0.0219	0.0211	**0.0208**	**0.0208**	0.0212
	H = 0.3	0.0021	0.0061	0.0040	0.0031	0.0027	0.0024	0.0023	0.0022	**0.0021**	**0.0021**
	H = 0.5	$1.0*10^{-4}$	$5.6*10^{-4}$	$2.9*10^{-4}$	$2.0*10^{-4}$	$1.6*10^{-4}$	$1.4*10^{-4}$	$1.2*10^{-4}$	$1.1*10^{-4}$	$1.1*10^{-4}$	**$1.0*10^{-4}$**
	H = 0.7	$3.3*10^{-6}$	$4.9*10^{-5}$	$1.9*10^{-5}$	$1.1*10^{-5}$	$7.6*10^{-6}$	$5.8*10^{-6}$	$4.8*10^{-6}$	$4.2*10^{-6}$	$3.7*10^{-6}$	**$3.5*10^{-6}$**
	H = 0.9	$7.0*10^{-8}$	$2.6*10^{-6}$	$8.1*10^{-7}$	$4.1*10^{-7}$	$2.6*10^{-7}$	$1.8*10^{-7}$	$1.3*10^{-7}$	$1.1*10^{-7}$	$9.0*10^{-8}$	**$8.0*10^{-8}$**
Internet data	—	0.0286	0.0310	**0.0289**	**0.0289**	0.0297	0.0310	0.0326	0.0345	0.0370	0.0399

FBM, fractional Brownian motion.

and $L(k) = (\bar{\lambda}(k), \bar{\lambda}(k-1), \ldots, \bar{\lambda}(1))'$; it is clear that the normalized EWMA estimator can be simply written as $\hat{\lambda}(k) = A'(k)L(k)$. Denote the covariance matrix of $\bar{\lambda}(1), \ldots, \bar{\lambda}(k)$ by

$$\Gamma(k) = \begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{k-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{k-1} & \gamma_{k-2} & \cdots & \gamma_0 \end{bmatrix}, \tag{7}$$

where $\gamma_0 = \sigma^2 = \mathrm{Var}(\bar{\lambda}(j))$, $j = 1, \ldots, k$. Therefore, the normalized EWMA estimator has the properties that $E\left[\hat{\lambda}(k)\right] = \lambda$ and $\mathrm{Var}(\hat{\lambda}(k)) = A'(k)\Gamma(k)A(k)$. On the basis of this equation, the control limits of an EWMA chart can be constructed as follows: The upper and lower control limits (UCL/LCL) of the k-th observation are set to

$$\mathrm{UCL/LCL}(k) = \lambda \pm c\sqrt{A'(k)\Gamma(k)A(k)}. \tag{8}$$

In practice, λ is replaced by the estimate $\bar{\bar{\lambda}}(k)$ (the mean of $\bar{\lambda}(1), \ldots, \bar{\lambda}(k)$), and $\Gamma(k)$ is replaced by the estimate $\hat{\Gamma}(k)$, wherein γ_j can be estimated by

$$\hat{\gamma}_j = \frac{1}{k-j}\sum_{i=1}^{k-j}\left[\bar{\lambda}(i+j) - \bar{\bar{\lambda}}(k)\right]\left[\bar{\lambda}(i) - \bar{\bar{\lambda}}(k)\right],\ j = 1, \ldots, k-1. \tag{9}$$

It is clear that if the process mean is known, then $\hat{\gamma}_j$ in (9) is a natural unbiased estimator of γ_j (i.e., the expected value of $\hat{\gamma}_j$ is γ_j). However, for mathematical simplicity, the denominator $k - j$ in (9) is sometimes replaced by k. With such modification, it becomes a consistent estimator of γ_j (i.e., converges in probability to γ_j as the number of observations goes to infinity) [37]. Note that these properties can be obtained without placing additional structural assumptions on the original process. For stronger theoretical results, the readers can refer to [38, 39] and the references therein.

On the basis of all the estimated quantities, the EWMA chart generates an out-of-control signal at time k if $\hat{\lambda}(k) > \mathrm{UCL}(k)$ or $\hat{\lambda}(k) < \mathrm{LCL}(k)$, and $c > 0$ is often a constant chosen by the user.

Adaptive choice of c

It is known that if the value of c is chosen to be large, then the process will generate fewer out-of-control signals; whereas if the value of c is chosen to be small, then the process will generate more out-of-control signals. Because the underlying process and statistics can possibly change, choosing a fixed value of c may overproduce or underproduce the out-of-control signals, thus resulting either high operating costs or bad performance. We next introduce a procedure that allows us to adaptively choose the value of c so that some preset operating cost will not be exceeded.

Because every out-of-control signal requires recalculation of the optimal control and thus a switching between control mechanisms, a natural idea is to preset an upper bound for the switching frequency (i.e. switching cost). Suppose the system has a limited operating cost (this is often the real situation) so that the switching frequency is not allowed to exceed a value $p < 1$. This is then equivalent to presetting a lower bound $\frac{1}{p}$ for the so-called *average run length* (ARL) of the control chart (i.e. the average length that an out-of-control signal is generated). Thus, the value of c should be flexibly adjusted so that the ARL has at least the value $\frac{1}{p}$. To avoid producing arbitrarily small control limits (so that the control chart becomes too sensitive), we also setup a lower bound for the value of c, say, let $c \geqslant c_0$. The guideline for how to adjust the value of c is shown as follows.

Guideline for adjusting c : Let $c_0 > 0$ be the initial value of c (e.g., we can always choose $c_0 = 0.5$). When m out-of-control signals are generated, calculate the ARL and denote it by $\widehat{\mathrm{ARL}}$. Adjust the value of c using

$$c^* = \max\left\{c_0, c \cdot \frac{F^{-1}\left(1 - \frac{p}{2}\right) - \hat{\lambda}(k)}{F^{-1}\left(1 - \frac{1}{2\widehat{\mathrm{ARL}}}\right) - \hat{\lambda}(k)}\right\}. \tag{10}$$

Note that in (10), F represents the *empirical distribution* of all obtained $\hat{\lambda}(j)$, and we assume that the distribution is fairly symmetrical. Thus, if $\hat{\lambda}(k)$ is obtained such that m out-of-control signals are generated, it is expected that

$$\hat{\lambda}(k) + c\sqrt{A'(k)\hat{\Gamma}(k)A(k)} \approx F^{-1}\left(1 - \frac{1}{2\widehat{\mathrm{ARL}}}\right). \tag{11}$$

Because we wish to adjust the value of c by c^* such that

$$\hat{\lambda}(k) + c^* \sqrt{A'(k)\hat{\Gamma}(k)A(k)} \approx F^{-1}\left(1 - \frac{p}{2}\right), \tag{12}$$

Equation (10) is then obtained by combining (11) and (12). As we can see from (10), to approach the upper bound of the switching frequency, the value of c is increased when $\widehat{ARL} < \frac{1}{p}$, whereas the value of c is decreased when $\widehat{ARL} > \frac{1}{p}$. To illustrate the implementation of this guideline, suppose the empirical distribution of $\hat{\lambda}(j)$ reveals to be (approximately) normal, the formulas for adjusting the value of c can be written as

$$c^* = \max\left\{ c_0, c \cdot \frac{\Phi^{-1}\left(1 - \frac{p}{2}\right)}{\Phi^{-1}\left(1 - \frac{1}{2\widehat{\text{ARL}}}\right)} \right\}, \tag{13}$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution.

Remark 2
If the empirical distribution F appears to be asymmetric, one can either (i) transform the data to possibly yield a symmetric distribution (e.g., the Box–Cox transformation may work) or (ii) modify the control chart by using asymmetric control limits [40, 41]. For (ii), the control limits in (8) can be presented as

$$\text{UCL}(k) = \lambda + c_U \sqrt{A'(k)\Gamma(k)A(k)} \ \text{ and } \ \text{LCL}(k) = \lambda - c_L \sqrt{A'(k)\Gamma(k)A(k)}, \tag{14}$$

where $c_U \neq c_L$. Therefore, by choosing an initial value $c_U = c_L = c_0 > 0$, the values of c_U and c_L can be adjusted using

$$c_U^* = \max\left\{ c_0, c_U \cdot \frac{F^{-1}\left(1 - \frac{p}{2}\right) - \hat{\lambda}(k)}{F^{-1}\left(1 - \frac{1}{2\widehat{\text{ARL}}}\right) - \hat{\lambda}(k)} \right\} \tag{15}$$

and

$$c_L^* = \max\left\{ c_0, c_L \cdot \frac{F^{-1}\left(\frac{p}{2}\right) - \hat{\lambda}(k)}{F^{-1}\left(\frac{1}{2\widehat{\text{ARL}}}\right) - \hat{\lambda}(k)} \right\}, \tag{16}$$

respectively.

To illustrate the proposed normalized EWMA estimator and the corresponding control chart, we consider a particular real traffic trace collected from the Indianapolis and Cleveland links of the Abilene backbone network. For simplicity, the trace was transformed into a time sequence of correlated arrival events as expressed in 100 kB per 1.5 s (see http://moat.nlanr.net/Images/Nvabar/ 2_MNA.html and Rolls *et al.* [36] for a complete description of this data set). The result of choosing $c_0 = 0.5$, with the value of c being adjusted by (13), is shown in Figure 2.

Note that in Figure 2, the lower bound (and initial value) of c is chosen to be $c_0 = 0.5$, which is a rule of thumb we obtained from a large number of simulation trials. In addition, the upper bound of the switching frequency is given by $p = 0.01$, and the number of out-of-control signals used to estimate the ARL is $m = 3$. As we can see from Figure 2, in the beginning of the first period (from time 0 to 2000), a few more out-of-control signals are generated for a relatively small value of c, whereas as time moves on, the value of c is automatically adjusted (increased) so that fewer out-of-control signals are generated to approach the preset frequency upper bound 0.01. In addition, the out-of-control signals were promptly generated right after time 2000, 4000, and 6000, where the intensity appears to have significant changes. For comparison purposes, we perform a simple sensitivity analysis on the choice of the lower bound c_0. The results for choosing $c_0 = 0.3$ and $c_0 = 0.7$ are shown in Figures 3 and 4, respectively.

As shown in Figure 3, a few more out-of-control signals (particularly in the beginning) are generated by decreasing the lower bound c_0 to 0.3. The result is straightforward because the control chart becomes more sensitive on detecting small changes of the intensity. On the other hand, the control chart becomes less sensitive on detecting small changes by increasing the lower bound c_0 to 0.7. As expected, fewer out-of-control signals are generated (see Figure 4). In practice, the choice of c_0 is subject to the sensitivity of control chart the user would like to maintain. However, it should be noted that for both cases, the value of c is automatically adjusted by (13) so that the frequency of out-of-control signals will approach the same upper bound 0.01.

We next consider the general Q-input system and describe how the proposed EWMA control charts are used to decide when to resolve the optimal control.

2.3. *Framework implementation*

We first assume that all input traffic flows are mutually independent; the detailed steps of how to implement the proposed framework and its integration with the control policy are summarized as follows.

Algorithm for framework implementation.

Step 1.
Choose T, c_0, p, m, and an arbitrary initial control $u^* = u_0$; set $k = 2$.
Record all subsequent arriving jobs over the time interval $(0, T]$ for each queue q and compute the traffic intensity $\bar{\lambda}_q(1)$.
Set the normalized EWMA estimator $\hat{\lambda}_q(1) = \bar{\lambda}_q(1)$ (i.e., set $\beta(1) = 1$) for each queue q.
Solve the optimal control u^* on the basis of $\hat{\Lambda}(1) = (\hat{\lambda}_1(1), \ldots, \hat{\lambda}_Q(1))$ and implement it at time T.
Set

$$\mathrm{LCL}_q(1) = \bar{\lambda}_q(1) - c_0\hat{\sigma}_q \ \text{ and } \ \mathrm{UCL}_q(1) = \bar{\lambda}_q(1) + c_0\hat{\sigma}_q \ \text{ for all } q.$$

Step 2.
Record all subsequent arriving jobs over the time interval $((k-1)T, kT]$ for all queues.
For each queue q, compute the cumulative average intensity $\bar{\bar{\lambda}}_q(k)$, estimate $\hat{\rho}_1(k)$, and obtain the normalized EWMA estimator $\hat{\lambda}_q(k)$.

Step 3.
If $\mathrm{LCL}_q(k) \leqslant \hat{\lambda}_q(k) \leqslant \mathrm{UCL}_q(k)$ for all inputs, then the optimal control u^* does not change.
Otherwise, if an out-of-control signal is generated at some input q, resolve u^* on the basis of $\hat{\Lambda}(k) = (\hat{\lambda}_1(k), \ldots, \hat{\lambda}_Q(k))$ and update the control policy at time kT.
If there are exactly m out-of-control signals generated, adjust the value of c using (10); otherwise, retain the value of c.
Reset the control limits $\mathrm{LCL}_q(k)$ and $\mathrm{UCL}_q(k)$ using (8).
Set $k = k + 1$, go back to Step 2.

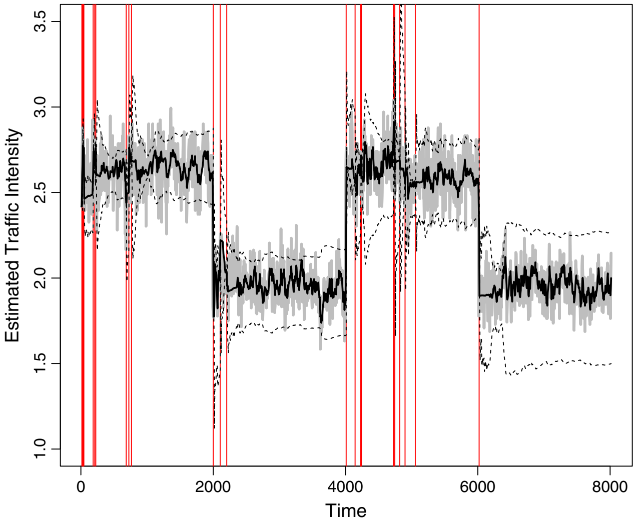

Figure 2. An illustration of the proposed normalized exponentially weighted moving average estimator (black solid line) and the corresponding control limits (dashed lines) for the data (gray line) collected from the Abilene backbone network. Note that here $c_0 = 0.5$, c is adjusted by (13), and the vertical lines represent the times when the out-of-control signals are generated.

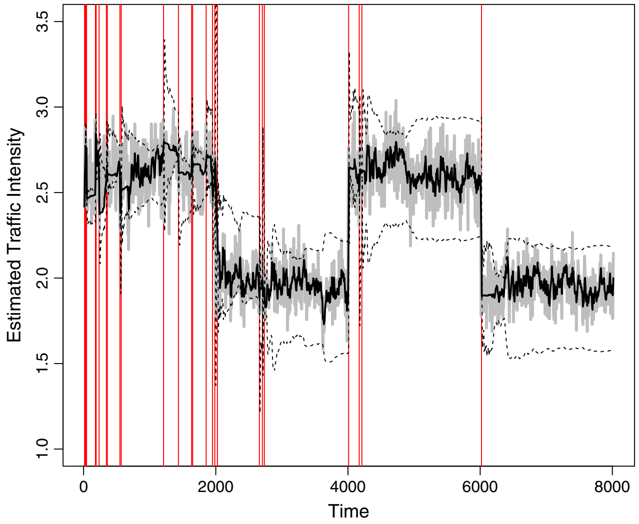

Figure 3. An illustration of the proposed normalized exponentially weighted moving average estimator (black solid line) and the corresponding control limits (dashed lines) for the data (gray line) collected from the Abilene backbone network. Note that here $c_0 = 0.3$, c is adjusted by (13), and the vertical lines represent the times when the out-of-control signals are generated.

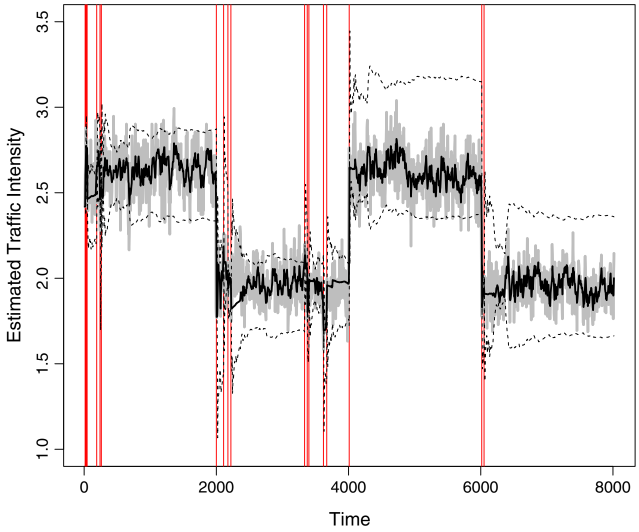

Figure 4. An illustration of the proposed normalized exponentially weighted moving average estimator (black solid line) and the corresponding control limits (dashed lines) for the data (gray line) collected from the Abilene backbone network. Note that here $c_0 = 0.7$, c is adjusted by (13), and the vertical lines represent the times when the out-of-control signals are generated.

Note that in Step 1, $\hat{\sigma}_q$ is the estimate of the standard deviation of $\bar{\lambda}_q(1)$, which can be calculated by tracking the observed data for each input traffic trace. If a simulation study is carried out to examine the long-term performance of the proposed framework, repeating Step 2 and Step 3 in the algorithm is desirable. For general Q-input systems where the traffic flows are inter-correlated, an extension is to use multivariate EWMA (MEWMA) control charts, which deal with multiple variates simultaneously. Assume that the Q-estimated traffic intensities over the kth time interval (in vector notation $\bar{\Lambda}(k)$) have the mean $\Lambda(k)$ and covariance matrix $\Sigma(k)$; the MEWMA statistic (in vector notation) is given by

$$\hat{\Lambda}(k) = B\bar{\Lambda}(k) + (I - B)\hat{\Lambda}(k-1) \tag{17}$$

where $B = \text{diag}(\beta_1, \ldots, \beta_Q)$, $0 < \beta_q \leqslant 1$, $q = 1, \ldots, Q$, and I is the $Q \times Q$ identity matrix. Therefore, the covariance matrix of $\hat{\Lambda}(k)$ can be written as a form of $f(\Sigma(k))$, where function f incorporates the intra-flow covariance into each of the inter-flow covariance in $\Sigma(k)$ (this requires a certain amount of algebra). The MEWMA chart then gives an out-of-control signal if

$$T^2(k) = \left(\hat{\Lambda}(k) - \bar{\bar{\Lambda}}(k)\right)' f(\Sigma(k))^{-1} \left(\hat{\Lambda}(k) - \bar{\bar{\Lambda}}(k)\right) > h, \tag{18}$$

where $\bar{\bar{\Lambda}}(k)$ is the vector of cumulative average traffic intensities and h can be (i) chosen to achieve a specified in-control ARL (Lowry and Woodall [42]) or (ii) adaptively adjusted in a similar fashion to the value of c.

Remark 3
It is noted that the estimation and monitoring of traffic intensity introduced previously is conducted by integrating all observed data over a preselected time period. However, in some applications where a large number of events can occur over a small time period (e.g. high speed internet), the proposed framework may not be practical because of limited storage space or computation resources. To overcome this problem, one can employ a suitable sampling technique so that a good estimation of traffic intensity can be obtained by utilizing merely partial information of data. For possible sampling techniques, the readers can refer to [43].

3. Performance assessment

In this section, we first introduce an optimal fluid control problem for the queueing systems described in Section 1. We next assess the performance of the optimal control policy (based on the proposed framework) on a number of systems with various types of input traffic.

3.1. An optimal fluid control problem for queueing systems

Let us recall the queueing systems introduced in Section 1. Let $\mathcal{I}_q = (t_j^q, \delta_j^q)$ denote the input process of jobs of queue q, where $t_j^q \in \mathbb{R}_+$ is the arrival time of the jth job and δ_j^q is the corresponding service requirement, $j \in \mathbb{Z}_+$, $q = 1, \ldots, Q$. It is noted that here we do not require strong structural assumptions on the arrival processes (such as independent and identically distributed inter-arrival times and service requirements); instead, any dependence between inter-arrival times, service requirements, and input traffic flows is allowed. There are M service modes, and at any point in time t, each service mode m can be possibly employed with a level $u_m(t)$, $0 \leqslant u_m(t) \leqslant 1$ for all $m = 1, \ldots, M$, and $\sum_{m=1}^{M} u_m(t) \leqslant 1$. When the mth mode is employed at 100% level (i.e., $u_m(t) = 1$), the jobs in queue q receive service at rate μ_{mq} (i.e., the amount of work that can be processed in one time unit). Therefore, mode m is associated with the service rate vector $R_m = (\mu_{m1}, \ldots, \mu_{mQ})$. All the available vectors R_m can be collected in the $M \times Q$ *service rate matrix* R, to which the zero service mode $R_0 = \vec{0}$ is appended that captures the idling periods in the system. We also assume that service rates are $additive$; that is, if different service modes are employed simultaneously at time t, they pool their efforts together. Therefore, the pooled service rate for each queue q at time t is denoted by $\sum_{m=1}^{M} u_m(t)\mu_{mq}$.

Note that any control policy π coincides with a particular process $u(t) = (u_1(t), \ldots, u_M(t)) \in \mathbb{R}_+^M$, and we assume it is non-anticipative. Let $W_q(t)$ denote the amount of work accumulated in queue q at time t; its evolution equation is then given by

$$W_q(t) = W_q(0) + \sum_{j \in \mathbb{Z}_+} \delta_j^q \mathbf{1}_{\{t_j^q \in (0,t]\}} - \sum_{m=1}^{M} \mu_{mq} \int_0^t u_m(s) \mathbf{1}_{\{W_q(s)>0\}} \mathrm{d}s \tag{19}$$

for every $q = 1, 2, \ldots, Q$. For any given time T, define the expected work coming into each queue by

$$\mathbb{E} \sum_{j \in \mathbb{Z}_+} \delta_j^q \mathbf{1}_{\{t_j^q \in (0,T]\}} = \int_0^T \lambda_q(t)\mathrm{d}t \tag{20}$$

where $\lambda_q(t)$ represents the *stochastic traffic intensity* of queue q at time t. Let $\Lambda(t) = (\lambda_1(t), \ldots, \lambda_Q(t))$, and we further assume that $0 < \int_0^T \lambda_q(t)\mathrm{d}t < \infty$ for any finite planning time horizon $[0, T]$, $q = 1, \ldots, Q$. The *long-term traffic intensity* of queue q is then given by

$$\lambda_q = \lim_{T \to \infty} \left[\frac{1}{T} \int_0^T \lambda_q(t)\mathrm{d}t \right], \quad q = 1, \ldots, Q. \tag{21}$$

It should be noted that any naive policy would not utilize the resources (i.e. $u(t)$) appropriately so as to optimize the performance measure(s) of interest. Therefore, finding the optimal control policy becomes a much harder task.

Fluid approximations have proved to be an important tool to study the performance of control policies in steady state [8,9,44]. The idea is to replace the workload by a continuous fluid level and where discrete job processes are approximated by a continuous fluid flow. Such a formulation can lead to deterministic systems (e.g., constant input and service rates) of coupled differential equations in the time variable that are solvable by conventional numerical methods. Consider the rescaled workload process $\bar{W}_q(t) = \lim_{k\to\infty} \frac{1}{k} W_q(kt)$; then, the evolution of the workload process (in fluid sense) can be written as

$$\bar{W}_q(t) = \bar{W}_q(0) + \int_0^t \lambda_q(s)\mathrm{d}s - \sum_{m=1}^{M} \mu_{mq} \int_0^t u_m(s)\mathrm{d}s, \; q = 1,2,\ldots,Q, \; t \geqslant 0, \tag{22}$$

with $\bar{W}_q(t) \geqslant 0$. Under mild assumptions, the aforementioned model is differentiable in t, and we obtain

$$\dot{\bar{W}}_q(t) = \frac{\mathrm{d}\bar{W}_q(t)}{\mathrm{d}t} = \lambda_q(t) - \sum_{m=1}^{M} u_m(t)\mu_{mq}, \tag{23}$$

which eliminates lower order stochastic fluctuations but is still influenced by the stochastic intensity $\lambda_q(t)$. For the whole system, we can write

$$\frac{\mathrm{d}\bar{W}(t)}{\mathrm{d}t} = \Lambda(t) - \sum_{m=1}^{M} u_m(t)R_m = \Lambda(t) - u(t)R, \tag{24}$$

where the operation between vectors is considered component-wise.

We focus next on a fixed time horizon $[0, T]$ and eliminate transient dynamics. Suppose that all time-dependent quantities are frozen except for the workload process $\bar{W}(t)$ and the corresponding queue length process $\bar{N}(t) = (\bar{N}_1(t), \ldots, \bar{N}_Q(t))$; that is, it is assumed that $\Lambda(t) \equiv \Lambda$ and $u(t) = u$ or $\vec{0}$. Hence, we obtain $\frac{d\bar{W}(t)}{dt} = \Lambda - uR$ for all $t \in [0, T]$, and the flow balance equations are obtained by setting $d\bar{W}(t)/dt = 0$, that is, $\Lambda = uR$.

Suppose now each queue has an initial fluid level proportional to its input rate; that is, $\bar{W}_q(0) = \lambda_q T$ for some $T > 0$, $q = 1, \ldots, Q$. For any deterministic control process u, denote the average service rate devoted to queue q by $(uR)_q$ (the qth element in uR) and assume that $\lambda_q \leqslant (uR)_q$ for all $q = 1, \ldots, Q$ (i.e., $d\bar{W}(t)/dt < 0$). The quantity

$$\frac{\lambda_q T}{(uR)_q - \lambda_q} \tag{25}$$

then represents the average amount of time it takes for the fluid present in queue q to be drained. Letting $C_q > 0$ denote the cost of holding the fluid in queue q per unit of time, we get that an optimal control process u^* would correspond to the solution of the following optimization problem:

$$\begin{array}{ll} \underset{u}{\text{minimize}} & \max\limits_{q} C_q\left(\dfrac{\lambda_q}{(uR)_q - \lambda_q}\right) \\ \text{subject to} & \Lambda \leqslant uR \end{array} \tag{26}$$

where vector inequalities are considered component-wise.

It is noted that the solution of (26) can be translated to the optimal control that minimizes (approximately) the maximal average holding cost (of workload) among all queues in steady state. This can be viewed as an extension of the analogous result for the M/M/1 queue, by an application of Little's law and assuming that all the input processes have exponential service requirements [45]. Therefore, the optimal control process u^* can potentially improve the QoS performance measures, such as delay (i.e., the waiting time for the job until service is first provided). To integrate the proposed framework with the control formulation in (26), each intensity λ_q is replaced by its estimate $\hat{\lambda}_q(t)$, whereas the optimal control u^* is recalculated only if large shifts of $\hat{\lambda}_q(t)$ are detected.

Remark 4

Equation (26) can be easily transformed into a linear programming (LP) problem by considering the reciprocal of the destination function. Thus, efficient algorithms provided by many statistical/mathematical software packages (such as MATLAB) can be utilized to solve the optimal control. Moreover, when the costs of holding one unit of work for all queues are the same (i.e. C_q are the same for all q), the optimal control u^* can be easily obtained by considering a dual problem.

Remark 5
In some cases where the constraint $\Lambda \leqslant uR$ in (26) is not satisfied (e.g., the input has 'burst' of job arrivals over a small time interval), the optimal control u^* can be solved by considering an alternative optimization problem:

$$\underset{u}{\text{minimize}} \ \max_q C_q \frac{\lambda_q}{(uR)_q}. \tag{27}$$

3.2. A benchmark control policy: MaxProduct

To assess the performance of the control policy obtained from our proposed framework, here we introduce a class of well-known control policies for this particular queueing system, called 'MaxProduct'. It is noted that for comparison purposes, this control policy will serve as a benchmark in the simulation study presented later.

The MaxProduct policies, proposed by Armony and Bambos [2], were constructed as follows. For the workload process $\vec{W}(t) = (w_1(t), \ldots, w_Q(t))$ and each service rate vector R_m, define their weighted inner product by

$$< \vec{W}(t), R_m >_{\vec{\alpha}} = \sum_{q=1}^{Q} \alpha_q w_q(t) \mu_{mq}, \tag{28}$$

where $\alpha_q > 0$ is an arbitrarily placed weight for jobs in queue q, $q = 1, \ldots, Q$. The MaxProduct policy corresponds to the control process $u(t) = (u_1(t), \ldots, u_M(t))$, where each component $u_m(t)$ is determined by

$$u_m(t) = \begin{cases} 1 & \text{if } m = \arg\max_{m' \in \{1,\ldots,M\}} < \vec{W}(t), R_{m'} >_{\vec{\alpha}}, \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

Note that the MaxProduct policy allows merely one service mode to be used at any point in time (i.e. service modes can not be shared simultaneously). If more than one service mode achieves the maximal inner product, for example, $u_i(t) = u_j(t) = 1$ for some $i \neq j$, the policy will randomly select one service mode from them.

It was shown by Armony and Bambos [2] that the MaxProduct policy can maximize the system's throughput for any arbitrarily chosen positive weight vector $\vec{\alpha}$. However, because $\vec{\alpha}$ affects the policy only through its directions in $\mathbb{R}_+^Q$ [1], it suffices to look at a particular class of policies for which the weight vectors satisfy that $\sum_{q=1}^{Q} \alpha_q = 1$, or equivalently $\sum_{q=1}^{Q-1} \alpha_q \leqslant 1 \left(\text{where } \alpha_Q = 1 - \sum_{q=1}^{Q-1} \alpha_q\right)$. For simplicity, in the rest of this study, the class of MaxProduct polices is denoted by $\pi_{\vec{\alpha}}$, whereas our proposed policy is denoted by π_{LP}. The systems examined are as follows: (i) a two-queue system fed by FBM input traffic with changing rates; (ii) a three-queue system fed by compound Poisson input processes with changing rates; and (iii) an eight-queue system fed by real network traces collected from the Abilene backbone network. Two performance metrics considered are the average system delay and the 95th percentile of job delays. Further, the frequencies of service-mode switchings were recorded and compared over a specific simulation time horizon. It is noted that for the two-queue and three-queue systems, we also include the simulation results of two ideal $oracle$ policies. The first policy, denoted by Oracle-I, knows the exact time of input-rate changes, whereas the second policy, denoted by Oracle-II, knows the exact input traffic process (i.e. both the input rates and the time of input-rate changes).

3.3. A two-queue system with fractional Brownian input traffic

Consider a two-queue system having three service modes $R_1 = (0, 4)$, $R_2 = (2, 3)$, and $R_3 = (3, 0)$. We next introduce the following fractional Brownian type of input traffic, considered by Leland *et al.* [46] and Norros [47, 48]:

$$W_q(t) = \lambda_q t + \sigma_q Z_q(t), \quad q = 1, 2, \tag{30}$$

where $W_q(t)$ is the total service requirement accumulated in the time interval $(0, t]$, $Z_1(t)$ and $Z_2(t)$ are independent *normalized FBMs* with the same Hurst parameter H, λ_q is the traffic intensity, and σ_q is the variance of traffic in a time unit. For simplicity purposes, we assume that the job inter-arrival times of each queue are deterministic, say, equal to one unit. Further, assume that the system starts at time 0 and that the nth job of queue q carries the service requirement,

$$W_q(n) - W_q(n-1) = \lambda_q + \sigma_q [Z_q(n) - Z_q(n-1)], \tag{31}$$

$q = 1, 2$. Assume that the system starts at time 0 with the pair of traffic intensities (λ_1, λ_2) changes periodically at certain point in time, say, $t = \{2500, 5000, 7500, \cdots\}$. Specifically, define the traffic intensities at time t by $(\lambda_1(t), \lambda_2(t)) = (0.2, 3.8)$ for $t \in [2500i, 2500(i+1))$, $i = 0, 4, 8, \ldots$; $(\lambda_1(t), \lambda_2(t)) = (1.9, 2.9)$ for $t \in [2500i, 2500(i+1))$, $i = 1, 5, 9, \ldots$; $(\lambda_1(t), \lambda_2(t)) = (2.5, 1.4)$ for $t \in [2500i, 2500(i+1))$, $i = 2, 6, 10, \ldots$; and $(\lambda_1(t), \lambda_2(t)) = (2.8, 0.2)$ for $t \in [2500i, 2500(i+1))$, $i = 3, 7, 11, \ldots$. We also assume that the variance functions for both queues are $\sigma_1(t) = 10$, $t \in [2500i, 2500(i+1))$, $i = 0, 2, 4, 6, \ldots$, and $\sigma_2(t) = 50$, $t \in [2500i, 2500(i+1))$, $i = 1, 3, 5, 7, \ldots$, respectively. Note that such choices for λ_q and σ_q can guarantee that $W_q(t)$ will not go negative, almost surely. In addition, the Hurst parameter is chosen to be $H = 0.6$ for $Z_q(t)$, so that the two input processes are characterized as *long-range dependent*. The resulting delay metrics for the class of MaxProduct policies (with different choices of queue weights $\vec{\alpha}$) are shown in Figure 5. The numerical results show that the minimal average system delay is given around $\vec{\alpha}* = (0.45, 0.55)$, whereas the minimal 95th percentile of delays is given around $\vec{\alpha}* = (0.6, 0.4)$.

The detailed delay performance and the operating costs of our proposed policy (with $p = 0.2$ and 0.33, $T = 5$, $c_0 = 0.5$, $m = 20$, and c being adjusted by (13)), the MaxProduct policy (with the optimal choice of queue weights $\vec{\alpha}*$), and the two oracle policies are given next in Table II.

As can be seen from Table II, our proposed policy (π_{LP}) with both switching frequency upper bounds ($p = 0.20$ and 0.33) outperforms the entire class of MaxProduct policies $\pi_{\vec{\alpha}}$ in terms of both the average system delay and the 95th percentile of delays. In particular, it performs better when the control chart is allowed to generate more out-of-control signals (i.e., when the preset upper bound of switching frequency is larger). For such a small system, it has a dramatic reduction on both the average system delay (94%) and the 95th percentile of delays (98%) compared with the MaxProduct policy with the optimal choice of queue weights (in practice, we even do not know how to choose them). In addition, it has less than (nearly) two orders of magnitude smaller service-mode switching frequency compared with the MaxProduct policies for the two possible choices of upper bounds p. This is because (i) by setting an upper bound, the proposed framework

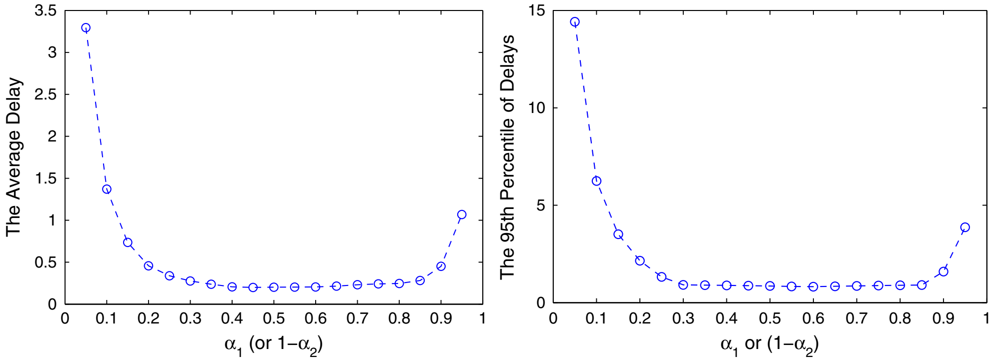

Figure 5. (Left panel): The resulting average system delays under the MaxProduct policy with all possible choices of queue weights $\vec{\alpha} = (\alpha_1, \alpha_2)$. (Right panel): The resulting 95th percentiles of delays under the MaxProduct policy with all possible choices of queue weights $\vec{\alpha} = (\alpha_1, \alpha_2)$.

Table II. The delay performance and operating costs of the control policies under consideration. Here $\pi_{\vec{\alpha}*}$ denotes the MaxProduct policy with the optimal choice of queue weights.

Control policy	Average delay	The 95th percentile of delays	Switching frequency for 5 time units	Main computational complexity
π_{LP} $p = 0.33$	0.012	0.020	0.3265	Estimate subsequent traffic intensities
$p = 0.20$	0.015	0.027	0.1870	Solving subsequent LP
$\pi_{\vec{\alpha}*}$	0.199	0.823	9.9998	Computing inner product sorting
Oracle-I	18.024	154.262	0.0020	Solving subsequent LP
Oracle-II	34.809	278.980	0.0020	Solving subsequent LP

LP, linear programming.

forced the control charts not to generate too many out-of-control signals; (ii) the underlying traffic intensities are long-range dependent and change fairly slow; and (iii) the service requirements have small variances. However, the nature of the MaxProduct policy is such that despite the lack of significant changes in $\hat{\Lambda}(k)$, the policy still resulted in many mode switchings. In summary, our proposed policy has a relatively small operating cost in terms of frequency of service-mode switching and exhibits a pretty good overall delay performance by choosing a fairly small value of p.

It is also noted that by choosing a fairly small p, our proposed policy significantly outperforms the two oracle policies in terms of both delay metrics. The result is not surprising here because for such long-range dependent processes with a fairly slow change of input rates, the oracle policies can not perform well because of the lack of response to local traffic fluctuations.

Remark 6

The number of service-mode switching for π_{LP} is the same as the number of out-of-control signal generated by the EWMA control chart used to monitor the changes of traffic intensity. On the other hand, the number of service-mode switching for the two oracle systems is the same as the exact number of changes for the underlying traffic intensity.

3.4. *A three-queue system with randomly modulated compound poisson input processes*

Consider a three-queue system having six service modes $R_1 = (0, 0, 6)$, $R_2 = (4, 0, 3)$, $R_3 = (6, 0, 0)$, $R_4 = (4, 5, 0)$, $R_5 = (0, 8, 0)$, and $R_6 = (0, 5, 3)$. The system is characterized by randomly modulated service modes that are defined by combinations of four possible input processes $\mathcal{I}_i = (\mathcal{I}_i^1, \mathcal{I}_i^2, \mathcal{I}_i^3,\ i = 1, 2, 3, 4)$. The combinations are mutually independent, and each $\mathcal{I}_i^q$ is a compound Poisson process with exponentially distributed service requirements. Further, each combination occurs with equal probability every 250,000 time units. Finally, the mean service requirements for the four combinations are as follows: $(\bar{\delta}_1^1, \bar{\delta}_1^2, \bar{\delta}_1^3) = (5.5, 0.25, 0.5)$, $(\bar{\delta}_2^1, \bar{\delta}_2^2, \bar{\delta}_2^3) = (0.28, 0.2, 5.6)$, $(\bar{\delta}_3^1, \bar{\delta}_3^2, \bar{\delta}_3^3) = (0.25, 7.5, 0.25)$, and $(\bar{\delta}_4^1, \bar{\delta}_4^2, \bar{\delta}_4^3) = (2.0, 2.3, 3.0)$, whereas for simplicity, the corresponding mean inter-arrival times are all chosen to be one. Therefore, the traffic intensities for these four input combinations are $\Lambda_1 = (5.5, 0.25, 0.5)$, $\Lambda_2 = (0.28, 0.2, 5.6)$, $\Lambda_3 = (0.25, 7.5, 0.25)$, and $\Lambda_4 = (2.0, 2.3, 3.0)$.

Computer simulations are performed to derive the average system delay and the 95th percentile of job delays for the policies under consideration. As discussed, for the class of MaxProduct policies $\pi_{\vec{\alpha}}$, it suffices to look at all pairs (α_1, α_2) such that $0 \leqslant \alpha_1, \alpha_2 \leqslant 1$ and $\alpha_1 + \alpha_2 \leqslant 1$ (i.e., a 2-simplex). The resulting contour plots of the average system delays and the 95th percentiles of job delays under policy $\pi_{\vec{\alpha}}$ with all possible choices of queue weights $\vec{\alpha}$ are shown in Figure 6. The numerical results show that the minimal average system delay is given around $\vec{\alpha}* = (0.35, 0.5, 0.15)$, whereas the minimal 95th percentile of delays is given around $\vec{\alpha}* = (0.3, 0.3, 0.4)$.

The detailed delay performance and the operating costs of our proposed policy (with $p = 0.2$ and 0.33, $T = 500$, $c_0 = 0.5$, $m = 20$, and c being adjusted by (13)), the MaxProduct policy (with optimal fixed queue weights $\vec{\alpha}*$), and the two oracle policies are given next in Table III. Analogously, the performance of our proposed policy becomes better by choosing a larger upper bound of switching frequency p for the EWMA control charts. In addition, it outperforms the entire class of the MaxProduct policies for both choices of p. It is noted that when $p = 0.33$, our policy has a significant improvement on both the average system delay (76%) and the 95th percentile of delays (75%) compared with the MaxProduct policy with the optimal choice of weights. Also, even for a rather large frequency limit (say, $p = 0.33$), our policy has over three orders of magnitude smaller number of service-mode switching than $\pi_{\vec{\alpha}}$. Note that compared with the systems with long-range dependent input traces (e.g., the FBM with $H = 0.6$ in the two-queue system), the two oracle

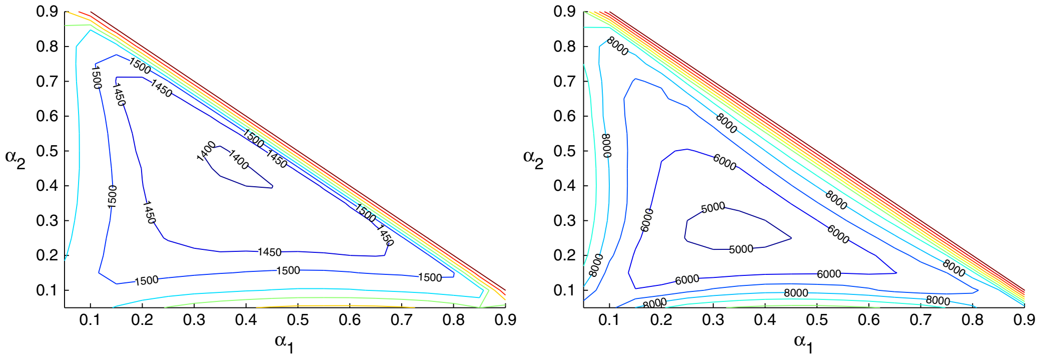

Figure 6. (Left panel): The resulting contour plot of the average system delay for $\pi_{\vec{\alpha}}$ with all possible choices of queue weights $\vec{\alpha}$. (Right panel): The resulting contour plot of the 95th percentile of job delays for $\pi_{\vec{\alpha}}$ with all possible choices of queue weights $\vec{\alpha}$.

Table III. The delay performance and operating costs of the control policies under consideration. Here $\pi_{\vec{\alpha}*}$ denotes the MaxProduct policy with the optimal choice of queue weights.

Control policy	Average delay	The 95th percentile of delays	Switching frequency for 500 time units	Main computational complexity
π_{LP} $p = 0.33$	338.01	1198.73	0.252	Estimate subsequent traffic intensities
$p = 0.20$	869.74	4062.21	0.137	Solving subsequent LP
$\pi_{\vec{\alpha}*}$	1396.03	4822.00	740.824	Computing inner product sorting
Oracle-I	1826.80	6139.12	0.002	Solving subsequent LP
Oracle-II	460.26	2754.51	0.002	Solving subsequent LP

LP, linear programming.

systems perform much better for these independent and identically distributed input flows. However, as expected, they do not perform particularly well because of a slow change of input rates.

3.5. *An eight-queue system with real traffic input traces*

We now investigate the performance of the proposed policy for larger systems. Consider an eight-queue system by feeding real network traces collected from the Indianapolis and Cleveland links of the Abilene backbone network. These traces were transformed into a sequence of correlated arrival events as expressed in 100 kB per 10 ms. In particular, eight input traces are extracted from the data source so that each input trace has 50,000 arrival events. As shown by Rolls *et al.* [36] and Hung and Michailidis [3], these traffic traces exhibit the property of self-similarity and long-range dependence. There are nine service modes R_1 =(5, 0, 0, 0, 0, 0, 0, 0), R_2 =(0, 6, 0, 0, 0, 0, 0, 0), R_3 =(0, 0, 6, 0, 0, 0, 0, 0), R_4 =(0, 0, 0, 7, 0, 0, 0, 0), R_5 =(0, 0, 0, 0, 5, 0, 0, 0), R_6 =(0, 0, 0, 0, 0, 6, 0, 0), R_7 =(0, 0, 0, 0, 0, 0, 7, 0), R_8 =(0, 0, 0, 0, 0, 0, 0, 7), and R_9 = (0.98, 0.8, 0.8, 0.69, 0.98, 0.8, 0.69, 0.69), where the service rate corresponds to the number of 100 kB that each service mode can process in 10 ms. Computer simulations are then performed to derive the average system delay and the 95th percentile of job delays under the control policies π_{LP} and $\pi_{\vec{\alpha}}$. Note that for such a system, $\pi_{\vec{\alpha}}$ has a considerable number of choices for the queue weight vector $\vec{\alpha}$ (e.g., all possible directions in $\mathbb{R}^8_+$). Because of limited simulation resources, we considered merely the 'standard' MaxProduct policy $\pi_{\vec{\alpha}=\vec{1}}$ (i.e., all queue weights are equally placed). The detailed delay performance and the operating costs of our proposed policy (with $p = 0.2$ and 0.33, $T = 1$, $c_0 = 0.5$, $m = 20$, and c being adjusted by (13)) and policy $\pi_{\vec{\alpha}=\vec{1}}$ are shown in Table IV.

As can be seen from Table IV, by considering fairly large limits for the switching frequency of service modes, our policy significantly outperforms the MaxProduct policy with equally placed queue weights, in terms of both the average system delay and the 95th percentile of job delays. Specifically, the improvement on both the average delay and the 95th percentile of job delays is about 37%. In addition, it requires less than 97% of service-mode switching in comparison with the standard MaxProduct policy.

As a general conclusion, there is a trade-off between performance with respect to delay and operating costs in terms of service-mode switching, as expected. However, the proposed control policy π_{LP} with appropriate switching frequency limits proves competitive across a large number of simulation scenarios. Further, as the number of queues increases,

Table IV. The delay performance and operating costs of policies π_{LP} and $\pi_{\vec{\alpha}=\vec{1}}$.

Control policy	Average delay	The 95th percentile of delays	Switching frequency for 1 time units	Main computational complexity
π_{LP} $p = 0.33$	251.148	620.501	0.2281	Estimate subsequent traffic intensities
$p = 0.20$	301.541	670.927	0.1753	Solving subsequent LP
$\pi_{\vec{\alpha}*}$	398.296	990.585	7.0388	Computing inner product sorting

LP, linear programming.

the cost of calculating all inner products, as required by the MaxProduct policies, proves prohibitive, which led Ross and Bambos [49, 50] to investigate alternatives on the basis of local searches.

Remark 7

The proposed method can be modified so that it is applicable for systems with customer abandonment (e.g., a call center with impatient customers or an inventory system with perishing products) [16, 51]. To do this, if a customer arrives at the system and abandons it in the current in-control period, then the up-to-date estimate $\hat{\lambda}(k)$ can be recalculated by excluding the impatient customer. This may or may not change the status from 'in-control' to 'out-of-control'. If a customer arrives at the system in some previous in-control period and abandons it in the current in-control period, then the up-to-date estimate $\hat{\lambda}(k)$ will not be changed (recall that the control policy is designed to be responsive to instantaneous traffic fluctuations). However, the performance of the proposed method for such systems needs to be further investigated.

4. Concluding remarks

In this paper, we propose a general framework for measuring and monitoring the flow intensities of stochastic systems. The framework employs an EWMA process to estimate the traffic intensity and monitor its 'significant' shifts over time. On the basis of the framework, an adaptive control policy can be easily developed so as to improve the performance measures of interest without requiring high operating costs. To illustrate the proposed framework, we introduce a fluid control problem for a canonical queueing model. Extensive simulation results show that the proposed framework exhibits a good performance in terms of both delay and operating costs for a number of scenarios. It is worth noting that the proposed framework is novel and superior to the existing methods from the following perspectives: (i) it is purely measurement based and adaptive to any types of input traffic flow; (ii) it can be applied to any stochastic systems having similar flow analogies and control schemes (i.e., systems wherein the control can be written as a function of flow intensities); and (iii) it is the first attempt to integrate the statistical online estimation, monitoring strategies, and operating costs into the control formulation of stochastic systems. We are currently investigating its performance when applied to the control problems for a variety of stochastic systems.

Acknowledgements

The authors would like to thank the two anonymous reviewers for their helpful comments. This work was supported by the research grant NSC 100-2118-M-004-005.

References

1. Hung YC, Michailidis G. A measurement based dynamic policy for switched processing systems. *Proceedings of IEEE International Conference on Communications*, Glasgow, Scotland, 2007; 301–306, DOI: 10.1109/ICC.2007.57.
2. Armony M, Bambos N. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems: Theory and Applications* 2003; **44**:209–252. DOI: 10.1023/A:1024714024248.
3. Hung YC, Michailidis G. Modeling, scheduling, and simulation of switched processing systems. *ACM Transactions on Modeling and Computer Simulation* 2008; **18**(3). DOI: 10.1145/1371574.1371578. Article 12.
4. Le Boudec JY. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory* 1998; **44**:1087–1096. DOI: 10.1109/18.669170.
5. Sorte DD, Reali G. Resource allocation rules for providing performance guarantees to traffic aggregates in a DiffServ environment. *Computer Communications* 2002; **25**:846–862. DOI: 10.1016/S0140-3664(01)00429-7.
6. Aktekin T, Soyer R. Bayesian analysis of queues with impatient customers: applications to call centers. *Naval Research Logistics* 2012; **59**:441–456. DOI: 10.1002/nav.21499.
7. Baccelli F, Bremaud P. *Elements of Queueing Theory*, 2nd edn. Springer: Berlin, Heidelberg, New York, 2003.
8. Dai JG. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability* 1995; **5**:49–77. DOI: 10.1214/aoap/1177004828.
9. Dai JG. *Stability of Fluid and Stochastic Processing Networks*. MaPhySto Miscellanea Publication, No. 9, Aarhus, 1999.
10. Harrison JM. The BIGSTEP approach to flow management in stochastic processing networks. *Stochastic Networks: Theory and Applications*, Kelly F, Zachary S, Ziendins I (eds). Oxford University Press: Oxford, 1996; 57–90.
11. Harrison JM. Heavy traffic analysis of a system with parallel servers: asymptotic optimality of discrete-review policies. *Annals of Applied Probability* 1998; **8**:822–848. DOI: 10.1214/aoap/1028903452.
12. Meyn SP, Tweedie RL. *Markov Chains and Stochastic Stability*. Springer-Verlag: London, 1993.
13. Meyn SP, Tweedie RL. Criteria for stability of Markovian processes III: Foster–Lyapunov criteria for continuous time processes. *Advances in Applied Probability* 1993; **25**:518–548. DOI: 10.2307/1427522.

14. Stolyar AL. MaxWeight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic. *Annals of Applied Probability* 2004; **14**:1–53. DOI: 10.1214/aoap/1075828046.
15. Wasserman KM, Michailidis G, Bambos N. Optimal processor allocation to differentiated job flows. *Performance Evaluation* 2006; **63**:1–14. DOI: 10.1016/j.peva.2004.11.001.
16. Zeltyn S, Mandelbaum A. Call centers with impatient customers: many-server asymptotics of the M/M/n+G queue. *Queuing Systems* 2005; **51**:36–402. DOI: 10.1007/s11134-005-3699-8.
17. Aktekin T, Soyer R. Call center arrival modeling: a Bayesian state-space approach. *Naval Research Logistics* 2011; **58**:28–42. DOI: 10.1002/nav.20436.
18. Weinberg J, Brown LD, Stroud JR. Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data. *Journal of the American Statistical Association* 2007; **102**:1185–1198. DOI: 10.1198/016214506000001455.
19. Asmussen S. Exponential families and regression in the Monte Carlo stud of queues and random walks. *Annals of Statistics* 1990; **18**:1851–1867. DOI: 10.1214/aos/1176347883.
20. Hung YC, Michailidis G, Bingham DR. Developing efficient simulation methodology for complex queueing networks. *Proceedings of the Winter Simulation Conference* 2003; **1**:512–519. DOI: 10.1109/WSC.2003.1261463.
21. Wieland JR, Pasupathy R, Schmeiser BW. Queueing-network stability: simulation-based checking. *Proceedings of the Winter Simulation Conference* 2003; **1**:520–527. DOI: 10.1109/WSC.2003.1261464.
22. Kleijnen JPC. Regression metamodels for simulation with common random numbers: Comparison of validation tests and confidence intervals. *Management Science* 1992; **38**:1164–1185. DOI: 10.1287/mnsc.38.8.1164.
23. Cheng RCH, Kleijnen JPC. Improved design of queueing simulation experiments with highly heteroscedastic responses. *Operations Research* 1999; **47**:762–777. DOI: 10.1287/opre.47.5.762.
24. Hayel Y, Ouarraou M, Tuffin B. Optimal measurement-based pricing for an M/M/1 queue. *Networks and Spatial Economics* 2007; **7**:177–195. DOI: 10.1007/s11067-006-9001-8.
25. Hung YC, Michailidis G. Improving quality of service for switched processing systems. *Proceedings of 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*, Trento, Italy, 2006; 46–53, DOI: 10.1109/CAMAD.2006.1649717.
26. Kallitsis MG, Michailidis G, Devetsikiotis M. Measurement-based optimal resource allocation for network services with pricing differentiation. *Performance Evaluation* 2009; **66**:505–523. DOI: 10.1016/j.peva.2009.03.003.
27. Xu P, Michailidis G, Devetsikiotis M. Profit-oriented resource allocation using online scheduling in flexible heterogeneous networks. *Telecommunication Systems* 2006; **31**:289–303. DOI: 10.1007/s11235-006-6525-7.
28. Egbelu PJ, Roy N. Material flow control in AGV/unit load based production lines. *International Journal of Production Research* 1988; **26**:81–94. DOI: 10.1080/00207548808947842.
29. Björklund E, Matinlauri I, Tierens A, Axelsson S, Forestier E, Jacobsson S, Ahlberg AJ, Kauric G, Mäntymaa P, Osnes L, Penttilä TL, Marquart H, Savolainen ER, Siitonen S, Torikka K, Mazur J, Porwit A. Quality control of flow cytometry data analysis for evaluation of minimal residual disease in bone marrow from acute leukemia patients during treatment. *Journal of Pediatric Hematology/Oncology* 2009; **31**:406–415. PMID: 19648789.
30. D'hautcourt JL. Quality control procedures for flow cytometric applications in the hematology laboratory. *Hematology and Cell Therapy* 1997; **38**:467–470. DOI: 10.1007/s00282-996-0467-0.
31. Seamer LC, Kuckuck F, Sklar LA. Sheath fluid control to permit stable flow in rapid mix flow cytometry. *Cytometry* 1999; **35**:75–79.
32. Bellizzi S, Bouc R. Adaptive sub-optimal parametric control for non-linear stochastic systems. Application to semi-active isolators. *Probability Methods in Applied Physics* 1995; **451**:223–238. DOI: 10.1007/3-540-60214-3_58.
33. Cox DR. Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society. Series B* 1961; **23**:414–442.
34. Montgomery DC, Mastrangelo CM. Some statistical process control methods for autocorrelated data. *Journal of Quality Technology* 1991; **23**:179–193.
35. Ramjee R, Crato N, Ray BK. A note on moving average forecasts of long memory processes with an application to quality control. *International Journal of Forecasting* 2002; **18**:291–297. DOI: 10.1016/S0169-2070(01)00159-5.
36. Rolls DA, Michailidis G, Hernandez-Campos F. Queueing analysis of network traffic: methodology and visualization tools. *Computer Networks* 2005; **48**:447–473. DOI: 10.1016/j.comnet.2004.11.016.
37. Percival DB. Three curious properties of the sample variance and autocovariance for stationary processes with unknown mean. *The American Statistician* 1993; **47**:274–276. DOI: 10.1080/00031305.1993.10475997.
38. Brockwell PJ, Davis RA. *Time Series: Theory and Methods*, Springer Series in Statistics. Springer-Verlag: New York, 1986.
39. Giurcanu M, Spokoiny V. Confidence estimation of the covariance function of stationary and locally stationary processes. *Statistics and Decisions* 2004; **22**:283–300. DOI: 10.1524/stnd.22.4.283.64315.
40. Chan LK, Heng JC. Skewness correction $\bar{X}$ and R charts for skewed distributions. *Naval Research Logistics* 2003; **50**:555–573. DOI: 10.1002/nav.10077.
41. Yi J, Prybutok VR, Clayton HR. ARL comparisons between neural network models and $\bar{x}$-control charts for quality characteristics that are nonnormally distributed. *Economic Quality Control* 2001; **16**:5–15. DOI: 10.1515/EQC.2001.5.
42. Lowry CA, Woodall WH. A multivariate exponentially weighted moving average control chart. *Technometrics* 1992; **34**:46–53. DOI: 10.2307/1269551.
43. Duffield N. Sampling for passive internet measurement: a review. *Statistical Science* 2004; **19**:472–498. DOI: 10.1214/088342304000000206.
44. Dai JG, Meyn SP. Stability and convergence of moments for multi-class queueing networks via fluid limit models. *IEEE Transactions on Automatic Control* 1994; **40**:1889–1904. DOI: 10.1109/9.471210.
45. Walrand J. *Introduction to Queueing Networks*. Prentice Hall: Englewood Cliffs, NJ, 1988.
46. Leland WE, Taqqu MS, Willinger W, Wilson DV. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking* 1994; **2**:1–15. DOI: 10.1109/90.282603.
47. Norros I. A storage model with self-similar input. *Queueing Systems: Theory and Applications* 1994; **16**:387–396. DOI: 10.1007/BF01158964.

48. Norros I. On the use of fractional Brownian motion in the theory of connectionless networks. *IEEE Journal on Selected Areas in Communications* 1995; **13**:953–962. DOI: 10.1109/49.400651.
49. Ross K, Bambos N. Local search scheduling algorithms for maximal throughput in packet switches. *Proceedings of IEEE INFOCOM* 2004; **2**:1158–1169. DOI: 10.1109/INFCOM.2004.1357002.
50. Ross K, Bambos N. Dynamic quality of service control in packet switch scheduling. *Proceedings of IEEE International Conference on Communications* 2005; **1**:396–401. DOI: 10.1109/ICC.2005.1494382.
51. Bassamboo A, Randhawa RS. On the accuracy of fluid models for capacity sizing in queueing systems with impatient customers. *Operations Research* 2010; **58**:1398–1413. DOI: 10.1287/opre.1100.0815.