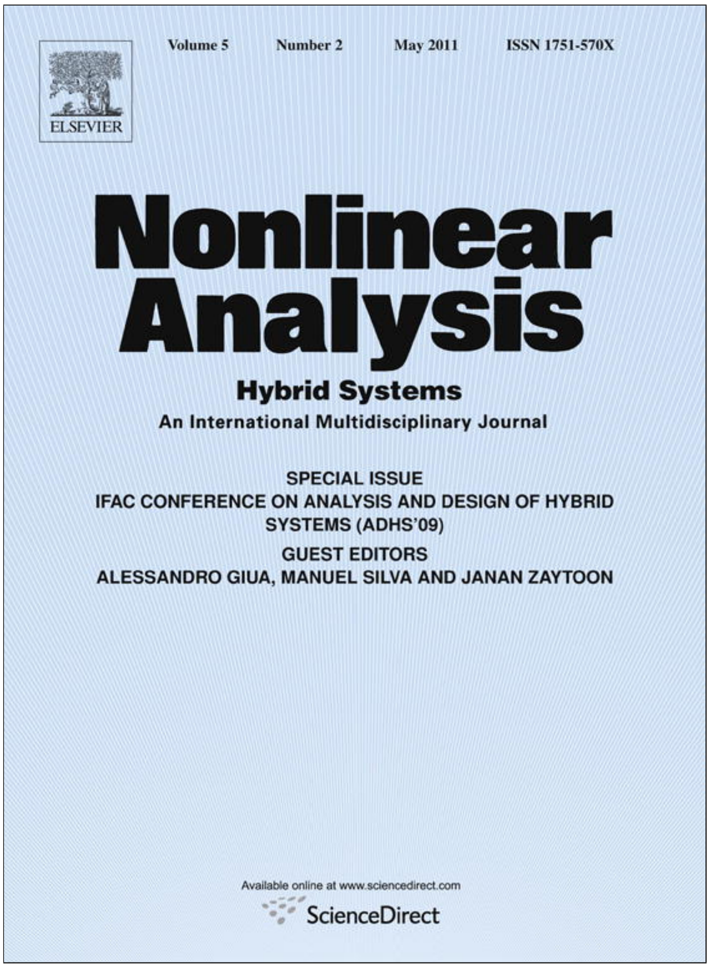

Volume 5 Number 2 May 2011 ISSN 1751-570X

ELSEVIER

Nonlinear Analysis

Hybrid Systems

An International Multidisciplinary Journal

SPECIAL ISSUE
IFAC CONFERENCE ON ANALYSIS AND DESIGN OF HYBRID SYSTEMS (ADHS'09)
GUEST EDITORS
ALESSANDRO GIUA, MANUEL SILVA AND JANAN ZAYTOON

Available online at www.sciencedirect.com
ScienceDirect

Nonlinear Analysis: Hybrid Systems 5 (2011) 301–319

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs

Resource contention games in multiclass stochastic flow models

Chen Yao, Christos Cassandras [*]

Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, United States

ARTICLE INFO

Article history:
Received 30 November 2009
Accepted 10 May 2010

ABSTRACT

We consider resource contention games in a stochastic hybrid system setting using Stochastic Flow Models (SFM) with multiple classes and class-dependent objectives. We present a general modeling framework for such games, where Infinitesimal Perturbation Analysis (IPA) estimators are derived for the derivatives of various class-dependent objectives. This allows us to study these games from the point of view of system-centric optimization of a performance metric and compare it to the user-centric approach where each user optimizes its own performance metric. We derive explicit solutions for a specific model in which the competing user classes employ threshold control policies and service is provided on a First Come First Serve (FCFS) basis. The unbiasedness of the IPA estimators is established in this case and it is shown that under certain conditions the system-centric and user-centric optimization solutions coincide.

1. Introduction

The study of hybrid systems is based on a combination of modeling frameworks originating in both time-driven and event-driven dynamic systems and resulting in *hybrid automata*. In a stochastic setting, such frameworks are augmented with models for random processes that affect either the time-driven dynamics or the events causing discrete state transitions or both. A general-purpose stochastic hybrid automaton model may be found in [1] along with various classes of *Stochastic Hybrid Systems* (SHS) which exhibit different properties or suit different types of applications. Stochastic Flow Models (SFMs) are examples of SHS obtained through an abstraction process applied to a large class of Discrete Event Systems (DES). They are especially useful in analyzing settings where users compete over different sharable resources, particularly communication networks with large traffic volumes (e.g., [2,3]). It should be stressed that such models may not always provide accurate representations for the purpose of analyzing the performance of the underlying system. What we are interested in, however, is control and optimization, in which case the value of a SFM lies in capturing only those system features needed to design an effective controller that can potentially optimize performance without any attempt at estimating the corresponding optimal performance value with accuracy. In particular, as explained next and explicitly seen in the analysis of subsequent sections, a SFM serves to provide only functional expressions for performance gradient estimators which are then used as surrogates of gradient estimates for the original DES. The values of these estimates, however, are still obtained based on information directly observed on the DES sample path, not the SFM whose sample paths are never used and never have to be realized.

While in most traditional fluid models the flow rates involved are treated as deterministic parameters, a SFM, as introduced in [4], treats flow rates as *stochastic processes*. With only minor technical assumptions imposed on the properties of such processes, a new approach for sensitivity analysis and optimization was recently proposed, based on Infinitesimal

[*] Corresponding author. Tel.: +1 617 353 7154; fax: +1 617 353 4830.
E-mail addresses: cyao@bu.edu (C. Yao), cgc@bu.edu (C. Cassandras).

1751-570X/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.nahs.2010.05.006

Perturbation Analysis (IPA) [5]. The essence of this approach is the on-line estimation of gradients (sensitivities) of certain performance measures with respect to various controllable parameters. These estimates may be incorporated in standard gradient-based algorithms to optimize system parameter settings. IPA was originally developed as a technique for evaluating gradients of sample performance functions in queueing systems and using them as unbiased gradient estimates of performance metrics expressed as expectations of these sample functions. However, IPA estimates become biased (hence unreliable for control purposes) when dealing with aspects of queueing systems such as multiple user classes, blocking due to limited resource capacities, and various forms of feedback control. The emergence of SFMs has rekindled the interest in IPA because SFMs allow us to circumvent these limitations, yielding simple unbiased gradient estimates of useful metrics even in the presence of blocking and a variety of feedback control mechanisms [6,7]. It is also possible to show [8] that for simple systems (e.g., G/G/1 queues with or without blocking) the IPA gradient estimators obtained through the SFM counterparts of those systems are the same as, or can asymptotically approximate, the gradient estimators obtained through IPA applied to the original DES.

When it comes to multiple user classes possibly competing for limited resources, IPA has been applied to problems where flows are differentiated in terms of admission to a system, but once admitted all flows are treated alike [9]. IPA for SFMs that can differentiate flow classes by associating different performance metrics to them has been a challenge and developing IPA estimates for gradients of class-dependent metrics has been elusive. Recently, [10] studied a multiclass SFM to analyze a dynamic priority call center. This model breaks new ground by differentiating among flow classes even after they enter the system; however, the analysis is very specific to the call center application and hard to extend to a general multiclass SFM. In addition, it is limited to state perturbations but not general performance metrics, and unbiasedness for the estimators derived was not established.

In [11], we developed a general multiclass SFM, where we introduced the class of "induced events" in addition to the two common event types (exogenous and endogenous) in previous SFMs. This new class of events greatly enriches the modeling power, but also considerably complicates IPA. Nonetheless, we developed IPA algorithms for estimating performance derivatives that subsequently allow us to optimize the underlying system. In [12], we extended our results and proposed a general IPA framework for stochastic hybrid systems with arbitrary structures. In the proposed multiclass SFM, each class is associated with its own performance metrics, such as workload, throughput, or loss rate due to overflow. This is an important new element in the analysis of SFMs, allowing us to study the difference between *user-centric* and *system-centric* optimization, something that was not previously possible, and to place resource contention problems in a *resource contention game* framework.

Resource contention games form a class of non-cooperative games in which two or more "users" compete for one or more sharable resources by submitting requests for their use over time. In problems studied thus far using SFMs, a purely "system-centric" point of view is adopted: the system defines an objective function and seeks to optimize it through appropriate control actions. In a resource contention game, however, there are multiple user types (referred to as user "classes") and each user defines its own objective function and seeks to optimize it. This gives rise to a game setting with a "user-centric" perspective. The contribution of this paper is to provide a general setting for resource contention games modeled through SFMs and then study a specific class of such games using IPA techniques to estimate user-specific performance derivatives and obtain both system-centric and user-centric solutions through gradient-based algorithms.

In Section 2 of the paper we present the general resource contention game setting using SFMs. We also describe how IPA can be used to estimate general-purpose performance metric derivatives with respect to parameters that the users control. In Section 3, we consider a specific class of games and present a detailed SFM which includes induced events to capture the user-dependent behavior of flows. We show how IPA is used to estimate performance metric derivatives for the specific problem and establish the unbiasedness of these estimates. We then compare user-centric to system-centric optimization for this problem and provide a condition under which the two solutions coincide. In Section 4, we include some numerical examples to illustrate the difference between the two solutions.

2. SFM framework for resource contention games

We consider a setting where N "players" corresponding to N user classes compete at time t for a resource with total service capacity $C(t)$, where $C(t)$ may be a random process. The ith user class submits requests at a rate $\alpha_i(t)$, generally time-varying and random. The ith user class is also allocated a portion of the service capacity $C(t)$ at time t, denoted by $c_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta},C(t))$, which satisfies for all $i=1,\ldots,N$:

$$\sum_{i=1}^{N} c_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta},C(t)) = C(t), \quad 0 \le c_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta},C(t)) \le C(t) \tag{1}$$

where $\mathbf{x}(t,\boldsymbol{\theta}) = [x_1(t,\boldsymbol{\theta}),\ldots,x_N(t,\boldsymbol{\theta})]$, $x_i(t,\boldsymbol{\theta}) \ge 0$ is the number of ith user class requests waiting to be processed at time t, and $\boldsymbol{\theta} = [\theta_1,\ldots,\theta_N]$ is a vector of control parameters used to determine how the resource capacity is allocated to different classes. Note that θ_i may itself be a vector of parameters all under the control of user class i. Also note that we use the notation $\mathbf{x}(t,\boldsymbol{\theta})$ to stress the dependence of the state on the parameter vector $\boldsymbol{\theta}$.

Adopting a SFM for this process, we let $x_i(t,\boldsymbol{\theta}) \in \mathbb{R}^+$ (where $\mathbb{R}^+$ denotes the set of non-negative real numbers) and observe that the system dynamics are given by

$$\frac{\mathrm{d}x_i(t,\boldsymbol{\theta})}{\mathrm{d}t^+} = f_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) = \begin{cases} 0 & x_i(t,\boldsymbol{\theta}) = 0 \text{ and } \alpha_i(t) - \beta_i(\mathbf{x}(\mathbf{t},\boldsymbol{\theta}),\boldsymbol{\theta}) \le 0 \\ \alpha_i(t) - \beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) & \text{otherwise} \end{cases} \tag{2}$$

where $\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$ is a control policy determining the actual service rate of class $i = 1,\ldots,N$ and satisfying for all t:

$$\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) \le c_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta},C(t)). \tag{3}$$

The random processes $\{C(t)\}$, $\{\mathbf{x}(t,\boldsymbol{\theta})\}$, and $\{\alpha_i(t)\}$, $\{\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})\}$ for all $i = 1,\ldots,N$ are defined on a common probability space $(\Omega,\mathcal{F},P)$. In addition, we will assume that $C(t)$ is independent of $\boldsymbol{\theta}$. The process $\{\alpha_i(t)\}$, on the other hand, may generally be dependent on $\boldsymbol{\theta}$.

Let $\mathcal{L}_i(\boldsymbol{\theta})$ be an objective function specified by the ith user class which, in general, may be defined over a finite interval $[0,T]$ or over an infinite horizon. The goal of the ith user class is to minimize

$$J_i(\boldsymbol{\theta}) = E\left[\mathcal{L}_i(\boldsymbol{\theta})\right]. \tag{4}$$

Clearly, the objective of user class i may conflict with that of another class $j \ne i$, giving rise to a non-cooperative resource contention game. On the other hand, from the system's perspective, we define an optimization problem which involves a weighted sum of all $J_i(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} w_i J_i(\boldsymbol{\theta}) \tag{5}$$

where $w_i \in \mathbb{R}^+$. The latter problem is termed "system-centric", while the former involves N separate optimization problems defining the "user-centric" approach. Under appropriate conditions, one can generally determine a solution $\boldsymbol{\theta}^s = [\theta_1^s,\ldots,\theta_N^s]$ to the system-centric optimization problem. The same is not always true for the user-centric approach if, for instance, θ_i and θ_j, $j \ne i$, are subject to some constraint of the form $g(\theta_i,\theta_j) \le 0$ that induces an interdependence between them. Even if the user-centric approach leads to a solution $\boldsymbol{\theta}^u = [\theta_1^u,\ldots,\theta_N^u]$, this, in general, differs from the system-centric solution $\boldsymbol{\theta}^s$. It is then interesting to compare, for each user class, $J_i(\boldsymbol{\theta}^s)$ and $J_i(\boldsymbol{\theta}^u)$ and seek ways to ensure that $J_i(\boldsymbol{\theta}^s) = J_i(\boldsymbol{\theta}^u)$ since we generally expect that $J_i(\boldsymbol{\theta}^s) < J_i(\boldsymbol{\theta}^u)$. The gap $J_i(\boldsymbol{\theta}^s) - J_i(\boldsymbol{\theta}^u) < 0$ is a measure of inefficiency of a user-centric approach and sometimes referred to as the "price of anarchy".

The dynamics in (2) reflect the hybrid nature of this system with each user class operating in at least two "modes" depending on the value of $x_i(t)$ and the sign of $\alpha_i(t) - \beta_i(\mathbf{x}(t),\boldsymbol{\theta})$. Additional modes are possible depending on the precise nature of the control policy $\beta_i(\mathbf{x}(t),\boldsymbol{\theta})$ as we will see in the next section. Using the notation established in [12] for general SHS, let us classify the events that may occur in the SFM defined by (1)–(3) as follows, where we shall use τ_k to denote the kth event time:

1. *Exogenous events*. An event is *exogenous* if its occurrence time τ_k is independent of the controllable vector $\boldsymbol{\theta}$, i.e., $\frac{\mathrm{d}\tau_k}{\mathrm{d}\boldsymbol{\theta}} = 0$. Exogenous events typically correspond to uncontrolled random changes in input processes. In our particular setting, the capacity $\{C(t)\}$ is a process independent of $\boldsymbol{\theta}$. Thus, if $\{C(t)\}$ is modeled as a piecewise constant process, then any event associated with a jump in $C(t)$ is, by definition, an exogenous event. The processes $\{\alpha_i(t)\}$ or $\{\beta_i(t)\}$ may also be independent of $\boldsymbol{\theta}$ for some or all $i = 1,\ldots,N$; if any such process is independent of $\boldsymbol{\theta}$, then additional exogenous events may similarly be defined.

2. *Endogenous events*. Using the definition in [12], an event occurring at time τ_k is *endogenous* if there exists a continuously differentiable function g_k such that

$$\tau_k = \min\{t > \tau_{k-1} : g_k(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) = 0\}. \tag{6}$$

Based on this definition, the event "$x_i(t,\boldsymbol{\theta})$ becomes 0" is an endogenous event with corresponding switching function

$$g(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) = x_i. \tag{7}$$

Additional endogenous events may be present depending on the specific nature of the control policies $\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$. If, for example, there is a discontinuity in $\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$ for $x_i(t) > 0$ that causes a switch in the time-driven dynamics $f_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$, then an endogenous event is defined with some associated switching function, as we will see in the next section.

3. *Induced events*. An event at time τ_k is *induced* if it is triggered by the occurrence of another event at time $\tau_m \le \tau_k$. The triggering event may be exogenous, endogenous, or itself an induced event. Induced events arise, for example, when it is necessary to model user requests on a First Come First Served (FCFS) basis as in [11], a case that we will revisit for the specific resource contention game defined in the next section.

In what follows, we study the processes through which event time and state derivatives (sensitivities) with respect to $\boldsymbol{\theta}$ evolve over time in a particular sample path without attempting to select any explicit control policies $\beta_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$ or

$c_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta},C(t))$. Using the IPA framework presented in [12], we define the following notation for all state and event time sample derivatives:

$$x'_{i,j}(t) \equiv \frac{\partial x_i(t)}{\partial \theta_j}, \qquad \tau'_{k,j} \equiv \frac{\partial \tau_k}{\partial \theta_j}, \quad i,j = 1,\ldots,N.$$

To simplify notation, we will write $\mathbf{x}(t)$ instead of $\mathbf{x}(t,\boldsymbol{\theta})$ when no ambiguity arises. Similarly, we write $f_i(t)$ instead of $f_i(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$.

Since $\mathbf{x}(t)$ is continuous at event times τ_k, i.e., $x_{i,j}(\tau_k^+) = x_{i,j}(\tau_k^-)$, taking derivatives on both sides with respect to θ_j gives

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) + \left[f_i(\tau_k^-) - f_i(\tau_k^+)\right] \cdot \tau'_{k,j}. \tag{8}$$

For any exogenous event at τ_k, by definition, $\tau'_{k,j} = \frac{\partial \tau_k}{\partial \theta_j} = 0$, therefore $x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-)$. On the other hand, for an endogenous event that satisfies $g_k(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta}) = 0$ in (6), taking derivatives with respect to θ_j gives

$$\frac{\partial g_k}{\partial \mathbf{x}}\left[\frac{\partial \mathbf{x}}{\partial \theta_j}(\tau_k^-) + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(\tau_k^-)\tau'_{k,j}\right] + \frac{\partial g_k}{\partial \theta_j} = 0$$

which implies that

$$\tau'_{k,j} = -\left[\sum_{i=1}^{N} \frac{\partial g_k}{\partial x_i} f_i(\tau_k^-)\right]^{-1} \left(\frac{\partial g_k}{\partial \theta_j} + \sum_{i=1}^{N} \frac{\partial g_k}{\partial x_i} x'_{i,j}(\tau_k^-)\right). \tag{9}$$

For the endogenous event "$x_i(t)$ becomes 0", we have $g_k(\mathbf{x},\boldsymbol{\theta}) = x_i$ and (9) reduces to

$$\tau'_{k,j} = -\frac{x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - \beta_i(\mathbf{x}(\tau_k^-),\boldsymbol{\theta})}. \tag{10}$$

Using this result in (8) and observing that $f_i(\tau_k^+) = 0$, we get

$$x'_{i,j}(\tau_k^+) = 0. \tag{11}$$

For any other endogenous event defined based on the specific form of $\beta_i(\mathbf{x}(t),\boldsymbol{\theta})$ when $x_i(t) > 0$, the event time derivatives $\tau'_{k,j}$ can be evaluated from (9), and then (8) can be used to obtain $x'_{i,j}(\tau_k^+)$.

If induced events are present, then, as shown in [12], the SFM must include additional state variables whose derivatives with respect to $\boldsymbol{\theta}$ must also be evaluated. The precise way in which induced events occur depends on the specific process we are interested in. We will return to this issue when analyzing the class of resource contention games considered in the next section.

We now return to the ith user class objective function $J_i(\boldsymbol{\theta}) = E\left[\mathcal{L}_i(\boldsymbol{\theta})\right]$. We are interested in the IPA derivative $\partial\mathcal{L}_i/\partial\theta_j$ which obviously depends on the specific form of $\mathcal{L}_i(\boldsymbol{\theta})$ defined by user class i. Here, for illustrative purposes, we limit ourselves to a common case where $\mathcal{L}_i(\boldsymbol{\theta})$ is defined as the average workload of class i over a given time interval $[0, T]$:

$$\mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{T}\int_0^T x_i(t)\mathrm{d}t$$

which can be re-written as

$$\mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{T}\sum_{k=0}^{N_T}\int_{\tau_k}^{\tau_{k+1}} x_i(t)\mathrm{d}t \tag{12}$$

where N_T is the number of events contained in $[0, T]$. In this case, the IPA derivative $\partial\mathcal{L}_i/\partial\theta_j$ can be obtained as

$$\frac{\partial\mathcal{L}_i}{\partial\theta_j} = \frac{1}{T}\sum_{k=0}^{N_T}\left[x_i(\tau_{k+1})\cdot\tau'_{k+1,j} - x_i(\tau_k)\cdot\tau'_{k,j} + \int_{\tau_k}^{\tau_{k+1}} x'_{i,j}(t)\mathrm{d}t\right] \tag{13}$$

where $\tau'_{k,j}$, $k = 1,\ldots,N_T$, are obtained as described above and $x'_{i,j}(t)$, $i,j = 1,\ldots,N$, are obtained by solving

$$\frac{\mathrm{d}}{\mathrm{d}t}x'_{i,j}(t) = \frac{\partial f_i}{\partial x_i}x'_{i,j}(t) + \frac{\partial f_i}{\partial \theta_j} \tag{14}$$

which is the result of taking derivatives on both sides of (2) with respect to θ_j. The solution of this equation is given by

$$x'_{i,j}(t) = \mathrm{e}^{\int_{\tau_k}^{t}\frac{\partial f_i(\mathbf{x}(u),\boldsymbol{\theta})}{\partial x_i}\mathrm{d}u}\left[\int_{\tau_k}^{t}\frac{\partial f_i(\mathbf{x}(v),\boldsymbol{\theta})}{\partial \theta_j}\cdot\mathrm{e}^{-\int_{\tau_k}^{v}\frac{\partial f_i(\mathbf{x}(u),\boldsymbol{\theta})}{\partial x_i}\mathrm{d}u}\mathrm{d}v + l\right] \tag{15}$$

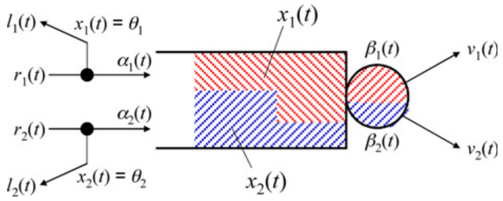

Fig. 1. A two-class stochastic flow model (SFM).

for $t \in [\tau_k, \tau_{k+1})$, where l is an initial condition obtained through (8) or similarly, depending on whether an additional endogenous or induced event is involved in the SFM. In addition, recalling the system dynamics in (2), $f_i(t) = 0$ during periods $[\tau_k, \tau_{k+1})$ when $x_i(t) = 0$; therefore, $\frac{\partial f_i}{\partial x_i} = \frac{\partial f_i}{\partial \theta_j} = 0$, then using (15), $x'_{i,j}(t) = x'_{i,j}(\tau_k^+) = 0$ according to (11) for all $t \in [\tau_k, \tau_{k+1})$ and (13) reduces to

$$\frac{\partial \mathcal{L}_i}{\partial \theta_j} = \frac{1}{T}\left[x_i(\tau_{N_T}) \cdot \tau'_{N_T,j} - x_i(\tau_0) \cdot \tau'_{0,j} + \sum_{k \in \Omega_i} \int_{\tau_k}^{\tau_{k+1}} x'_{i,j}(t)\mathrm{d}t\right] \tag{16}$$

where Ω_i is the set of all periods during which $x_i(t) > 0$, defined as

$$\Omega_i = \{k : x_i(t) > 0 \text{ for all } t \in [\tau_{k-1}, \tau_k],\ k = 1, \ldots, N_T\}.$$

Observe that the evaluation of $\mathrm{d}\mathcal{L}_i/\mathrm{d}\theta_j$ above only depends on information related to the event times τ_k and their derivatives and on $x'_{i,j}(t)$.

3. A class of resource contention games

In this section, we study a specific class of resource contention games where multiple users compete for access to a single sharable resource using threshold-based admission control and a FCFS processing policy. Each user defines a "class" of tasks that are randomly generated, placed in a common queue, and processed on a FCFS basis across all such classes. We first describe the SFM abstraction of such a system, limiting ourselves to two user classes (our analysis directly applies to three or more classes at the expense of added notation) and provide a formal stochastic hybrid automaton framework. Then, using IPA-based user-specific performance derivative estimates, we formulate a resource contention game and show that with the proper adjustment mechanism this user-centric approach can converge to the same optimal solution as that obtained through system-centric optimization.

3.1. SFM for the resource contention game

A two-class SFM is shown in Fig. 1. Similar to the general setting in the last section, we define several real-valued and non-negative random processes on a common probability space $(\Omega, \mathcal{F}, P)$. In particular, the process $\{x_i(t)\}$, $i = 1, 2$, defines the (real-valued) fluid content of class i in the system, $\{C(t)\}$ is the total service capacity at time t, and $\{c_i(t)\}$, $i = 1, 2$, characterizes the service capacities of both classes where, clearly, $C(t) = c_1(t) + c_2(t)$, consistent with (1). The external *arrival* flow processes $\{r_i(t)\}$, $i = 1, 2$, characterize the task arrival rates at time t. The parameter θ_i in this case is a threshold controlled by user class i to limit the inflow; alternatively, it may be viewed as a buffer capacity assigned to class i: When $x_i(t) \geq \theta_i$, some of the class i incoming flow is dropped, giving rise to the *overflow* or *loss* process $\{l_i(t)\}$ and the actual *input* flow process $\{\alpha_i(t)\}$, $i = 1, 2$, which are defined as follows:

$$\alpha_i(t) = \begin{cases} c_i(t) & x_i(t) = \theta_i \text{ and } r_i(t) \geq c_i(t) \\ r_i(t) & \text{otherwise} \end{cases} \tag{17}$$

$$l_i(t) = r_i(t) - \alpha_i(t) = \begin{cases} r_i(t) - c_i(t) & x_i(t) = \theta_i \text{ and } r_i(t) \geq c_i(t) \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Observe that both $\alpha_i(t)$ and $l_i(t)$ depend on θ_i. Similar to prior work on SFMs (e.g., [6,7]), the class i queue content can be either empty, full, or neither. Accordingly, an interval $[\tau_k, \tau_l]$, $k < l$, over which $x_i(t) = 0$ for all $t \in [\tau_k, \tau_l]$ corresponds to an *empty period* (EP) for this class, while an interval $[\tau_k, \tau_l]$, $k < l$, over which $x_i(t) = \theta_i$ for all $t \in [\tau_k, \tau_l]$ corresponds to a *full period*(FP). A *boundary* period (BP) is either an EP or FP; a *nonboundary* period (NBP) is a supremal interval during which $0 < x_i(t) < \theta_i$. Finally, the *output* flow processes $\{\beta_i(t)\}$, $i = 1, 2$, characterize departing flow rates, which in this case are specifically defined as

$$\beta_i(t) = \begin{cases} r_i(t) & x_i(t) = 0 \text{ and } r_i(t) \leq c_i(t) \\ c_i(t) & \text{otherwise.} \end{cases} \tag{19}$$

It is obvious from this definition that $\beta_i(t)$ satisfies condition (3). As illustrated in Fig. 1, the part of the content representing flow class i is generally time-varying, depending on θ_i or changes in $r_i(t)$ or $c_i(t)$; in the figure, the class 1 portion of the content decreases at some time instant, coinciding with some event which might, for instance, be a downward jump in $r_1(t)$.

We are interested in the behavior of this SFM over a finite time interval $[0, T]$. Regarding the arrival and service processes, we will impose no restrictions on their probabilistic characterizations, but will make the following assumption:

Assumption 3.1. W.p. 1, the arrival $r_i(t) \geq 0$, $i = 1, 2$, and service capacity $C(t) \geq 0$ are piecewise constant functions in the interval $[0, T]$.

We define a vector $\mathbf{x}(t) = [x_1(t), x_2(t)]$ as in the last section, and the system dynamics in (2) become:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t^+} = \begin{cases} 0 & x_i(t) = 0 \text{ and } r_i(t) \leq c_i(t) \\ 0 & x_i(t) = \theta_i \text{ and } r_i(t) \geq c_i(t) \\ \alpha_i(t) - \beta_i(t) & \text{otherwise.} \end{cases} \tag{20}$$

Thus, when $0 < x_i(t) < \theta_i$ we simply have $\frac{\mathrm{d}x_i}{\mathrm{d}t^+} = r_i(t) - c_i(t)$. When $x_i(t) = 0$ and $r_i(t) \leq c_i(t)$, the outflow rate is limited by the external arrival flow rate; similarly, when $x_i(t) = \theta_i$ and $r_i(t) \geq c_i(t)$, the inflow rate is limited by the service flow rate, leading to the loss rate $l_i(t)$ in (18). We will use $x(t) = \sum_{i=1}^{2} x_i(t)$ to denote the total system content at t.

The crucial difference between a single class SFM, as in [13], and the two-class SFM in Fig. 1 is the behavior of the service capacity $c_i(t)$. Whereas in the single-class model the service capacity allocation is independent of the system state, $c_i(t)$ in the two-class SFM depends on the queue contents and the inflow processes so as to satisfy the FCFS nature of the underlying DES as explained next. Initially, the service capacity is allocated proportional to the inflow rates, i.e.,

$$c_i(0) = C(0)\frac{\alpha_i(0)}{\sum\limits_j \alpha_j(0)}. \tag{21}$$

This allocation is maintained until there is a change in $\alpha_i(t)/\sum_j \alpha_i(t)$ at some time $t > 0$. When that happens, the total content $x(t)$ is the unprocessed workload under the initial service flow allocation. Let $\omega(t)$ denote the amount of time required to process this workload, at which point the new service rate allocation can take effect. Thus, the formal definition of $\omega(t)$ is through the relationship:

$$\begin{aligned} &\int_t^{t+\omega(t)} C(\tau)\mathrm{d}\tau = x(t) \quad x(t) > 0 \\ &\omega(t) = 0 \qquad x(t) = 0. \end{aligned} \tag{22}$$

Finally, at time $t + \omega(t)$ the new allocation takes effect:

$$c_i(t + \omega(t)) = C(t + \omega(t))\frac{\alpha_i(t)}{\sum\limits_j \alpha_j(t)}. \tag{23}$$

Therefore, in this SFM any event at t that causes a change in $\alpha_i(t)/\sum_j \alpha_i(t)$ is critical in that it "induces" another event at $t + \omega(t)$ which results in a service rate allocation change. Similarly to the general setting in Section 2, we use τ_k, $k = 1, 2 \ldots$, to denote event occurrence times in increasing order of k. Using this notation, we define the set of *inflow change events* in the interval $[0, \tau_k]$:

$$\Gamma_k = \{m : \exists i \in \{1, 2\} \text{ s.t. } \alpha_i(\tau_m^-) \neq \alpha_i(\tau_m^+),\ m \leq k\}. \tag{24}$$

To avoid degenerate cases where $C(\tau) = 0$ for all $\tau > t$, we will assume, whenever (22) is used, that $C(\tau) > 0$ for a sufficiently long time interval to ensure that $\omega(t) < \infty$.

The following lemma establishes the fact that the service flow allocation mechanism in (21)–(23) captures the FCFS nature of the underlying system modeled as a SFM (the proof of the lemma is given in [14]).

Lemma 3.1. *For $\omega(t)$ defined in (22),*

$$\int_t^{t+\omega(t)} c_1(s)\mathrm{d}s = x_1(t), \qquad \int_t^{t+\omega(t)} c_2(s)\mathrm{d}s = x_2(t).$$

This result asserts that any class i flow entering the SFM at t leaves at the same time $t + \omega(t)$ for both classes $i = 1, 2$. This is consistent with the defining property of a FCFS policy in a queueing system, i.e., the waiting time of a customer arriving at t in a FCFS queue is the same regardless of its class.

The presence of a delay $\omega(t)$ in (21)–(23) introduces *induced events* into this system. Based on the definition of induced events given in Section 2 (see also [12] and analysis therein), we need to augment our state description through state variables $y_m(t)$, $m = 1, 2, \ldots,$ associated with inflow change events occurring at times τ_m, $m = 1, 2 \ldots,$ to provide "timers" triggered when such an inflow change event occurs and then measure the amount of time until the queue content $x(\tau_m)$ is depleted, i.e.,

$$\frac{\mathrm{d}y_m(t)}{\mathrm{d}t} = \begin{cases} -C(t) & \tau_m \le t < \tau_m + \omega(\tau_m), \quad m \in \mathcal{F}_m \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$y_m(\tau_m^+) = \begin{cases} x(\tau_m) & y_m(\tau_m^-) = 0, \quad m \in \mathcal{F}_m \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, these state variables are only used for inflow change events, so that $y_m(t) = 0$ unless $m \in \mathcal{F}_m$. Intuitively, $y_m(t)$ decreases from $x(\tau_m)$ at the rate of the service capacity $C(t)$ until this queue content is depleted, at which time $y_m(\tau_m + \omega(\tau_m)) = 0$ and the associated induced event takes place.

This SFM includes all exogenous and endogenous events defined in the general setting of Section 2. In addition, there are: (i) endogenous events that initiate a FP for either user class with an associated switching function $g(\mathbf{x(t)}, \boldsymbol{\theta}) = x_i - \theta_i$, $i = 1, 2$, and (ii) induced events as described above caused by the service flow allocation mechanism in (21)–(23). We will refer to the latter as *ω-events*, because they are all related to the definition of $\omega(t)$ in (22). An event of this type occurring at time τ_k is "induced" by an inflow change event at some time $\tau_m < \tau_k$, that is, any event (exogenous, endogenous, or itself an ω-event) such that some $\alpha_i(t)$, $i = 1, 2$, changes value at $t = \tau_m$, $m \in \mathcal{F}_m$ as defined in (24). Thus, an event at time τ_k is an ω-event if there exists an event at τ_m, $m \in \mathcal{F}_m$, such that

$$\tau_k(\theta) = \tau_m + \omega(\tau_m) > \tau_m$$

$$\text{and} \quad \int_{\tau_m}^{\tau_m + \omega(\tau_m)} C(\tau)\mathrm{d}\tau = x(\tau_m) > 0.$$

It should be clear that an ω-event occurs at time $\tau_m + \omega(\tau_m)$ when the workflow $x(\tau_m)$ present at the time the event was induced becomes depleted and a service flow reallocation must result. If, however, $x(\tau_m) = 0$, by (22) we get $\omega(\tau_m) = 0$ and the event has no further effect on the SFM.

We stress again that an ω-event can occur in the following three ways: (i) Induced by an exogenous event at τ_m such that $\alpha_i(\tau_m^-) \neq \alpha_i(\tau_m^+)$. (ii) Induced by an endogenous event at τ_m. If $x_i(\theta, \tau_m) = 0$, then by (17) and (20) this is not an inflow change event. If, on the other hand, $x_i(\theta, \tau_m) = \theta_i$, from (17) and (20), we see that $\alpha_i(\tau_m^-) = r_i(\tau_m) > 0$ and $\alpha_i(\tau_m^+) = c_i(\tau_m^+) < r_i(\tau_m)$, i.e., this is an inflow change event. (iii) Induced by another ω-event at τ_m which takes place while $x_i(t) = \theta_i$, $i = 1, 2$. When this happens, (23) in conjunction with (17), causes a new inflow change at $\tau_k = \tau_m + \omega(\tau_m)$. In particular, (23) and (17) imply that:

$$\beta_i(\tau_k^+) = C(\tau_k)\frac{\alpha_i(\tau_m^+)}{\sum_j \alpha_j(\tau_m^+)}$$

$$\alpha_i(\tau_k^+) = \beta_i(\tau_k^+) = C(\tau_k)\frac{\alpha_i(\tau_m^+)}{\sum_j \alpha_j(\tau_m^+)}.$$

Thus, a chain of ω-events is generated. If we index the events forming this chain by s, these indices are a subset of $\{1, 2, \ldots\}$, and we get $\{\alpha_{i.s}\}$, $s = 1, 2, \ldots,$ with

$$\alpha_{i,s+1} = \frac{C(\tau_{s+1})}{\sum_j \alpha_{j,s}}\alpha_{i,s}.$$

The convergence properties of $\{\alpha_{i.s}\}$ depend on the ratio $C(\tau_{s+1})/\sum_j \alpha_{j,s}$. In contrast to a single-class SFM where no events occur during a FP, this sequence is potentially infinite if $x_i(t) = \theta_i$ for all $t > \tau_m$, which dramatically affects performance sensitivity in such resource contention settings. A complete analysis of $\{\alpha_{i,s}\}$ is provided in [14], but it is worth mentioning that if $C(\tau_{s+1})/\sum_j \alpha_{j,s} < 1$, which typically occurs when $r_i(t) \ge C(t)$ for some i, then $\{\alpha_{i.s}\}$ is monotonically decreasing with $\alpha_{i.s} \to 0$. Thus, class i may ultimately be denied access to service unless its FP ends before this limit is reached.

3.2. Stochastic hybrid automaton model

In this section, we develop a stochastic hybrid automaton model for the SFM in Fig. 1, which provides a formal setting facilitating its analysis. We begin by considering a typical sample path of the SFM in terms of the values $x_i(t)$ can take. We define discrete *aggregate states* by decomposing the sample path into intervals $(\pi, \pi']$, $\pi < \pi'$, that fall into one of three types: (i) $x_i(t) = 0$ for all $t \in (\pi, \pi']$, (ii) $0 < x_i(t) < \theta_i$ for all $t \in (\pi, \pi']$, and (iii) $x_i(t) = \theta_i$ for all $t \in (\pi, \pi']$. Denote

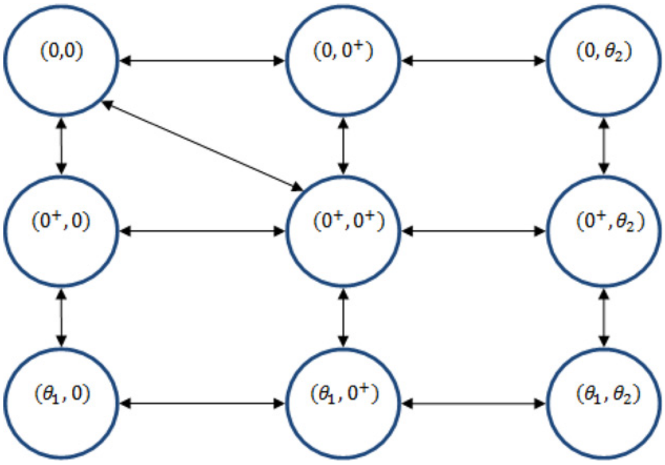

Fig. 2. Aggregate state transition diagram.

the associated aggregate state by $s_i(t)$ and the corresponding three values by 0, 0^+, and θ_i respectively. This defines a set $\Phi_i = \{0, 0^+, \theta_i\}$. It follows that for the two-class SFM we have an aggregate state $s(t) = (s_1(t), s_2(t))$ taking values in the set

$$\Phi = \{(0,0), (0,0^+), (0,\theta_2), (0^+,0), (0^+,0^+), (0^+,\theta_2), (\theta_1,0), (\theta_1,0^+), (\theta_1,\theta_2)\}.$$

Thus, an aggregate state value of 0 corresponds to an EP for this class, a value of θ_i corresponds to a FP, and aggregate state 0^+ corresponds to a NBP. The aggregate state transition diagram is shown in Fig. 2. Within each aggregate state, the system follows the dynamics (20) and all transitions are caused by events defined in the previous section. A distinctive feature in the diagram is the diagonal transition from $(0^+, 0^+)$ to $(0, 0)$, whose existence follows from Lemma 3.1 where we saw that any class i flow entering the SFM at t leaves at the same time $t + \omega(t)$ for both classes $i = 1, 2$. If $r_1(t) = r_2(t) = 0$ and $x_i(t) > 0$ for both $i = 1, 2$, then the lemma implies that the time to deplete the current content of each class is the same, i.e., both class contents become zero simultaneously, which corresponds to the transition from $(0^+, 0^+)$ to $(0, 0)$. In fact, this represents the most common form of queue depletion; the transitions from $(0^+, 0^+)$ to $(0, 0^+)$ or to $(0^+, 0)$ only occur by certain ω-events, as asserted by the following corollary (its proof is embedded in the proof of Lemma 3.1).

Corollary 3.1. *The transition from $(0^+, 0^+)$ to $(0, 0^+)$ must be caused by an ω-event triggered by an inflow change event that reduces $r_1 > 0$ to 0. Similarly, the transition from $(0^+, 0^+)$ to $(0^+, 0)$ must be caused by an ω-event triggered by an inflow change event that reduces $r_2 > 0$ to 0.*

Note that the hybrid automaton model in Fig. 2 is incomplete because it does not include the information captured by the state variables $y_m(t)$, $m = 1, 2, \ldots$, in (25). Suppose at time $t \in (\tau_k, \tau_{k+1}]$ the system is in the aggregate state (s_1, s_2) and transitions to (s_1', s_2') at τ_{k+1}. If the event at τ_{k+1} is an inflow change event, then from (25), y_{k+1} will become positive from 0; if the event at τ_{k+1} is an induced event, i.e., $\tau_{k+1} = \tau_m + \omega(\tau_m)$ for some m, then also based on (25), y_m will decrease to 0. In order to keep track of these "timers", we augment the aggregate state to $(s_1(t), s_2(t), n(t))$, where $n(t) = 0, 1, \ldots$, is the number of strictly positive y_k at t. From (25), $n(t)$ actually indicates the number of ω-events which have been induced by some inflow change event but have not yet occurred at time t. Clearly a complete transition diagram now becomes much more complicated. We limit ourselves to Fig. 3 showing all possible outgoing transitions from a typical state $(0^+, 0^+, n)$ with $n > 0$. For example, when the state is $(0^+, 0^+, n)$ and an inflow change event occurs, a new event is induced and the number of such events is increased by 1. Note that since $0 < x_i(t) < \theta_i$ for $i = 1, 2$ at $(0^+, 0^+, n)$, only an exogenous event (a jump in some $r_i(t)$) could induce a new future event. Similar diagrams can be obtained for other aggregate states. It is noteworthy in Fig. 3 that transitions from $(0^+, 0^+, \cdot)$ to $(0, 0^+, \cdot)$ or $(0^+, 0, \cdot)$ are caused by ω-events, consistent with the Corollary above.

3.3. Resource contention optimization

An optimization problem for the resource contention setting above is defined by treating $\boldsymbol{\theta} = [\theta_1, \theta_2]$ as a controllable parameter vector. Whereas in [11] we assumed that $\theta_i > 0, i = 1, 2$, are not otherwise constrained, in this paper we consider the case where

$$\theta_1 + \theta_2 = \Theta \tag{26}$$

i.e., a *fixed* amount of total queueing capacity is to be allocated to the two user classes. We seek to optimize performance metrics of the form $J(\boldsymbol{\theta}; x(0), T) = E[\mathcal{L}(\boldsymbol{\theta}; x(0), T)]$ where $\mathcal{L}(\boldsymbol{\theta}; x(0), T)$ is a sample function of interest evaluated in the interval $[0, T]$ with initial conditions $x(0)$. Typical performance metrics of interest are the loss volume (due to overflow

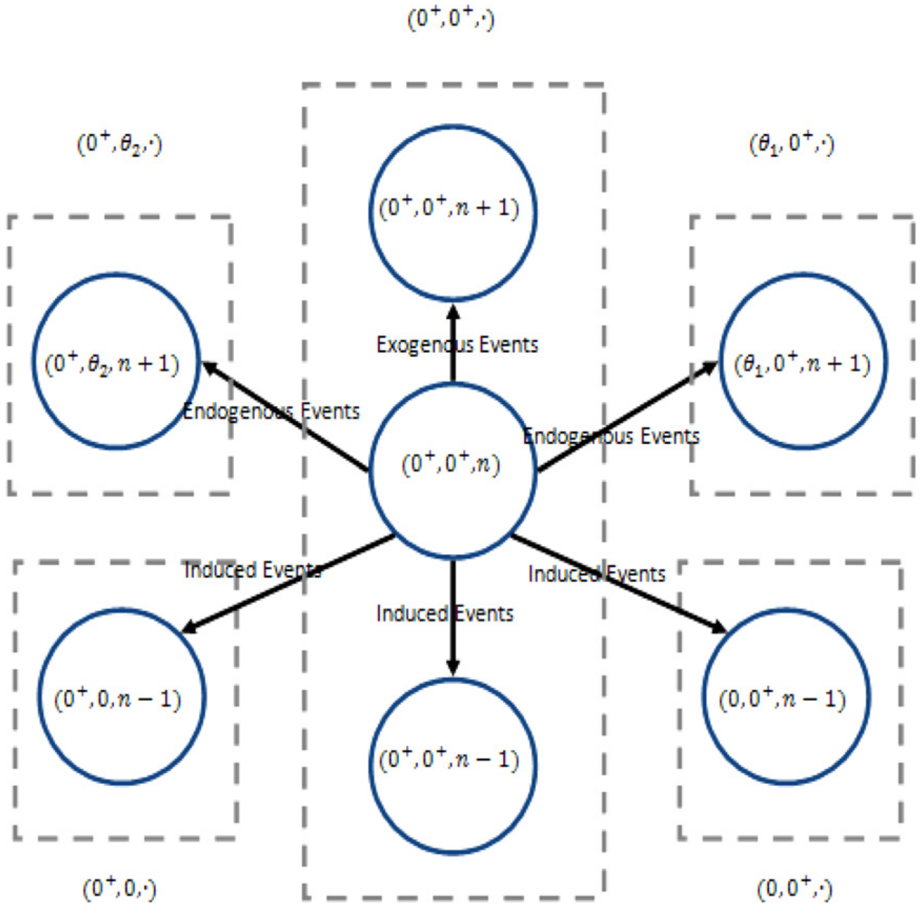

Fig. 3. Outgoing transitions from a state $(0^+, 0^+, n)$, $n > 0$, in the set $(0^+, 0^+, \cdot)$.

processes), the loss probability, the average workload (i.e., the queue contents), and the system throughput. In addition, delay metrics can be incorporated through fluid versions of Little's law. We shall limit ourselves to the class-dependent *loss volumes*, $L_i(\boldsymbol{\theta}; x(0), T)$, and *average workloads*, $Q_i(\boldsymbol{\theta}; x(0), T)$, $i = 1, 2$, and write for notational simplicity $L_i(\boldsymbol{\theta})$ and $Q_i(\boldsymbol{\theta})$. Let N_T be the total number of events observed in a sample path over $[0, T]$. The average workload of flow class i, $i = 1, 2$, is

$$Q_i(\boldsymbol{\theta}) = \frac{1}{T}\int_0^T x_i(t, \boldsymbol{\theta})\mathrm{d}t = \frac{1}{T}\sum_{k\in\Omega_i}\int_{\tau_{k-1}}^{\tau_k} x_i(t, \boldsymbol{\theta})\mathrm{d}t \tag{27}$$

with $\tau_0 = 0$, and Ω_i is the set of all *non-empty periods* (NEPs) for class i, defined as

$$\Omega_i = \{k : x_i(t) > 0 \text{ for all } t \in [\tau_{k-1}, \tau_k],\ k = 1, \ldots, N_T\}.$$

The average loss rate of flow class i, $i = 1, 2$, is

$$L_i(\boldsymbol{\theta}) = \frac{1}{T}\int_0^T l_i(t, \boldsymbol{\theta})\mathrm{d}t = \frac{1}{T}\sum_{k\in\Psi_i}\int_{\tau_{k-1}}^{\tau_k} [r_i(t) - \alpha_i(t)]\,\mathrm{d}t \tag{28}$$

where we have used the fact that $l_i(t) = r_i(t) - \alpha_i(t)$ and $l_i(t) \geq 0$ only in FPs of class i, with the definition:

$$\Psi_i = \{k : x_i(t) = \theta_i \text{ for all } t \in [\tau_{k-1}, \tau_k],\ k = 1, \ldots, N_T\}.$$

In the user-centric approach, class 1 and class 2 individually optimize their own performance metric defined as

$$J_i(\boldsymbol{\theta}) = \gamma_{1,i}E[Q_i(\boldsymbol{\theta})]\theta_i + \gamma_{2,i}E\left[L_i(\boldsymbol{\theta})\right] \tag{29}$$

where $\gamma_{i,j}$ are user-based weights reflecting the relative importance of workload and loss rate. We point out that although $J_i(\boldsymbol{\theta})$ is a function of both θ_1 and θ_2, class i is limited to controlling its own threshold parameter θ_i only. A game theoretic approach is a natural one in the context of the resource contention setting we have described, since users are generally not cooperative: rather, they compete for a limited resource (in our case, the queuing capacity Θ). Moreover, it is often the case that there is no centralized coordination and it is necessary to attain an equilibrium based on actions taken by the individual users without any central resource allocation control.

In the system-centric approach, we assume that there exists a central controller responsible for determining an optimal resource allocation based on a weighted sum of all classes' objective functions, as defined in (5), i.e.

$$J(\boldsymbol{\theta}) = w_1 J_1(\boldsymbol{\theta}) + w_2 J_2(\boldsymbol{\theta}). \tag{30}$$

Ideally, one would like to guarantee that the optimal solution obtained by the system-centric optimization process coincides with that of the user-centric approach, assuming that one even exists in this case. Thus, our goal will be to seek solutions for both optimization problems (if they exist) and study their difference.

Given that we do not wish to impose any limitations on the processes $\{r_i(t)\}$ and $\{C(t)\}$ (other than mild technical conditions), it is infeasible to obtain closed-form expressions for $J_i(\boldsymbol{\theta})$. Therefore, similar to our previous work in [11], we seek iterative gradient-based methods such as stochastic approximation algorithms (e.g., [15]) of the form

$$\theta_{i,k+1} = \theta_{i,k} - \eta_k H_{i,k}(\boldsymbol{\theta}_k; x(0), \xi_T), \quad k = 0, 1, \ldots \tag{31}$$

where, in the case of (30), $H_{i,k}(\boldsymbol{\theta}_k; x(0), \xi_T)$ is an estimate of $\partial J(\boldsymbol{\theta})/\partial\theta_i$, $i = 1, 2$, evaluated at $\boldsymbol{\theta} = (\theta_{1,k}, \theta_{2,k})$ over a sample path of length $[0, T]$ denoted by ξ_T. In the case of (29), $H_{i,k}(\boldsymbol{\theta}_k; x(0), \xi_T)$ is an estimate of $\partial J_i(\boldsymbol{\theta})/\partial\theta_i$, $i = 1, 2$, evaluated at $\boldsymbol{\theta} = (\theta_{1,k}, \theta_{2,k})$. We make use of IPA to obtain estimates of $\partial J(\boldsymbol{\theta})/\partial\theta_i$ or $\partial J_i(\boldsymbol{\theta})/\partial\theta_i$, which implies the need for evaluating the sample derivatives of $Q_i(\theta)$ and $L_i(\theta)$. It is clear from (27) and (28) that this requires the sample derivatives of the states $x_i(t, \boldsymbol{\theta})$ and of the event times $\tau_k(\boldsymbol{\theta})$ where the explicit dependence on the parameter $\boldsymbol{\theta}$ is included for emphasis.

3.4. IPA derivatives

Similar to the general setting in Section 2, we define the following for all state and event time sample derivatives:

$$x'_{i,j}(t) \equiv \frac{\partial x_i(t)}{\partial\theta_j}, \qquad y'_{m,j}(t) \equiv \frac{\partial y_m(t)}{\partial\theta_j}, \qquad \tau'_{k,j} \equiv \frac{\partial\tau_k}{\partial\theta_j} \tag{32}$$

for $i, j = 1, 2$, and $k, m = 1, 2, \ldots$. We will make use of the general expressions in (8), (9) and (15) to obtain $x'_{i,j}(t)$ and $\tau'_{k,j}$ for the specific SFM in Fig. 1. In addition, using the definition in (25), we can derive state and event time derivatives at induced events. Since an induced event at τ_k is triggered by an inflow change event at τ_m, such that $\tau_k = \tau_m + \omega(\tau_m)$, then, based on (25), we must have $y_m(\tau_k^-) = 0$, and taking derivatives on both sides with respect to θ_j we get:

$$y'_{m,j}(\tau_k^-) + \frac{\partial y_m(\tau_k^-)}{\partial t} \cdot \tau'_{k,j} = 0$$

which gives $y'_{m,j}(\tau_k^-) - C(\tau_k^-)\tau'_{k,j} = 0$ or (noting that $C(\tau_k^-) \neq 0$, otherwise depletion at τ_k is not possible):

$$\tau'_{k,j} = \frac{y'_{m,j}(\tau_k^-)}{C(\tau_k^-)}. \tag{33}$$

Along the same lines as the state derivatives $x'_{i,j}(t)$, we can also obtain $y'_{m,j}(t)$ as follows. First, between any two consecutive events,

$$y'_{m,j}(t) = y'_{m,j}(\tau_k^+) \quad t \in [\tau_k, \tau_{k+1}),\ m \in \mathcal{F}_k .$$

Since $y_m(t)$ is continuous as long as $y_m(t) > 0$ in (25), for any event that occurs while $y_m(t) > 0$,

$$y_m(\tau_k^+) = y_m(\tau_k^-), \quad m \in \mathcal{F}_{k-1} . \tag{34}$$

Taking derivatives with respect to θ_j,

$$y'_{m,j}(\tau_k^+) + \frac{\partial y_m(\tau_k^+)}{\partial t}\tau'_{k,j} = y'_{m,j}(\tau_k^-) + \frac{\partial y_m(\tau_k^-)}{\partial t}\tau'_{k,j}$$

where $\frac{\partial y_m(\tau_k^+)}{\partial t} = -C(\tau_k^+)$, $\frac{\partial y_m(\tau_k^-)}{\partial t} = -C(\tau_k^-)$ so that

$$y'_{m,j}(\tau_k^+) = y'_{m,j}(\tau_k^-) + [C(\tau_k^+) - C(\tau_k^-)]\tau'_{k,j}. \tag{35}$$

If $y_m(\tau_k^-) = 0$ and $m \in \mathcal{F}_{k-1}$, then, by definition, an induced event occurs at τ_k, and $\tau_k = \tau_m + \omega(\tau_m)$. Recalling (25), $y_m(t) = 0$ thereafter, so we also reset:

$$y'_{m,j}(\tau_k^+) = 0 \quad \text{if } \tau_k = \tau_m + \omega(\tau_m). \tag{36}$$

Finally, if an inflow change event occurs at τ_k, recall from (25) that at τ_k we initialize $y_k(t)$ by setting

$$y_k(\tau_k^+) = x(\tau_k^+) = x(\tau_k^-) = \sum_{i=1}^{2} x_i(\tau_k^-).$$

Therefore, taking derivatives with respect to θ_j,

$$y'_{k,j}(\tau_k^+) + \frac{\partial y_{k,j}(\tau_k^+)}{\partial t}\tau'_{k,j} = \sum_{i=1}^{2} x'_{i,j}(\tau_k^-) + \sum_{i=1}^{2} \frac{\partial x_i(\tau_k^-)}{\partial t}\tau'_{k,j}$$

and since $\frac{\partial y_{k,j}(\tau_k^+)}{\partial t} = -C(\tau_k^+)$, $\frac{\partial x_i(\tau_k^-)}{\partial t} = f_{i,k}(\tau_k^-)$, we get

$$y'_{k,j}(\tau_k^+) = \sum_{i=1}^{2} x'_{i,j}(\tau_k^-) + \left[\sum_{i=1}^{2} f_{i,k}(\tau_k^-) + C(\tau_k^+)\right]\tau'_{k,j}. \tag{37}$$

We can now derive the state derivatives $x'_{i,j}(t)$, $y'_{m,j}(t)$, and event time derivative $\tau'_{k,j}$ as evaluated at every event in the system. In between any two consecutive events, using (15) and noting that $\frac{\partial \beta_i}{\partial \theta_j} = 0$ for $i, j = 1, 2$, for all $t \in [\tau_k, \tau_{k+1})$ we have

$$x'_{i,j}(t) = x'_{i,j}(\tau_k^+), \qquad y'_{m,j}(t) = y'_{m,j}(\tau_k^+). \tag{38}$$

At event time τ_k, depending on the event type, we have:

1. *Exogenous events*. As already seen in Section 2, $\tau'_{k,j} = 0$ in this case, and using (8) and (34), we have

$$\tau'_{k,j} = 0, \qquad x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) \tag{39}$$
$$y'_{m,j}(\tau_k^+) = y'_{m,j}(\tau_k^-) \quad m \in \mathcal{F}_{k-1}\,.$$

2. *Endogenous events*. If an endogenous event initiates a FP for flow class i, then the switching function in (6) is $g(\mathbf{x}(\mathbf{t}), \boldsymbol{\theta}) = x_i - \theta_i$. Applying (9) and (8) gives, for $j = i$,

$$\tau'_{k,j} = \frac{1 - x'_{ij}(\tau_k^-)}{r_i(\tau_k^-) - c_i(\tau_k^-)}, \qquad x'_{ij}(\tau_k^+) = 1 \tag{40}$$

and for $j \neq i$,

$$\tau'_{k,j} = \frac{-x'_{i,j}(\tau_k^-)}{r_i(\tau_k^-) - c_i(\tau_k^-)}, \qquad x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-). \tag{41}$$

In addition, based on (36), $y'_m(\tau_k^+) = y'_m(\tau_k^-),\ m \in \mathcal{F}_{k-1}$.

If the event initiates an EP, we have already obtained in (11) and (10):

$$\tau'_k = \frac{-x'_{i,j}(\tau_k^-)}{r_i(\tau_k^-) - c_i(\tau_k^-)}, \qquad x'_{i,j}(\tau_k^+) = 0 \tag{42}$$

and based on (36), $y'_m(\tau_k^+) = y'_m(\tau_k^-)$, $m \in \mathcal{F}_{k-1}$.

3. *ω-events*. Suppose an ω-event occurs at τ_k induced by an inflow change event at τ_m so that $\tau_k = \tau_m + \omega(\tau_m)$. In this case, $f_{i,k}(\tau_k^-) = \alpha_i(\tau_k^-) - c_i(\tau_k^-)$, $f_{i,k+1}(\tau_k^+) = \alpha_i(\tau_k^+) - c_i(\tau_k^+)$, and based on (33) and (8),

$$\tau'_{k,j} = \frac{y'_{m,j}(\tau_k^-)}{C(\tau_k^-)} \tag{43}$$

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) + \left[\alpha_i(\tau_k^-) - \alpha_i(\tau_k^+) + c_i(\tau_k^+) - c_i(\tau_k^-)\right]\frac{y'_{m,j}(\tau_k^-)}{C(\tau_k^-)}. \tag{44}$$

In addition, using (35) and (36), we have

$$y'_m(\tau_k^+) = \begin{cases} 0 & \tau_k = \tau_m + \omega(\tau_m) \\ y'_m(\tau_k^-) & \text{otherwise} \end{cases}, \quad m \in \mathcal{F}_{k-1}\,.$$

Finally, if any event at τ_k is also an inflow change event, then we use (37) for the state variable y_k, where $\tau'_{k,j}$ is given by (39), (40), (41), (42), or (43) depending on the type of event that caused the inflow change.

Based on (39) through (44), we can evaluate all $x'_{i,j}(t)$ and $\tau'_{k,j}$ along a given sample path. We can then return to (27) and (28) and evaluate the performance metric derivatives:

$$\frac{\partial Q_i(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{T}\sum_{k \in \Omega_i} x'_{i,j}(\tau_{k-1}) \cdot (\tau_k - \tau_{k-1}) \tag{45}$$

$$\frac{\partial L_i(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{T}\sum_{k \in \Psi_i} \left[[r_i(\tau_{k-1}^+) - \alpha_i(\tau_{k-1}^+)](\tau'_{k,j} - \tau'_{k-1,j})\right]. \tag{46}$$

It is worth pointing out that a state derivative $x'_{i,j}(t)$ is always reset to zero when an EP for class i occurs as indicated in (42). This derivative can only become non-zero when a FP occurs at time τ_k, in which case $x'_{ii}(\tau_k^+) = 1$ as seen in (40). Subsequently, the value of $x'_{i,j}(t)$ may be modified through (41) and (44) until the next EP when it is reset to zero once again. It is also important to observe that the entire process of evaluating $\partial Q_i(\boldsymbol{\theta})/\partial\theta_j$ and $\partial L_i(\boldsymbol{\theta})/\partial\theta_j$ depends only on event time information and on flow rate values at certain event times. *The behavior of the SFM in between events does not affect the IPA process*. Thus, the optimization process is robust with respect to potential modeling errors in the time-driven dynamics.

The unbiasedness of the IPA derivatives $\partial Q_i(\boldsymbol{\theta})/\partial\theta_j$ and $\partial L_i(\boldsymbol{\theta})/\partial\theta_j$ can be ensured by Assumption 3.1 and the following additional assumption.

Assumption 3.2. (a) For every $\boldsymbol{\theta} \in \Theta$, w.p. 1, two events cannot occur at exactly the same time, unless one event induces the other, (b) W.p. 1, no two processes $\{r_i(t)\}$, $\{\beta_i(t)\}$, $i = 1, 2$, have identical values during any open subinterval of $[0, T]$.

We point out that even if the conditions of Assumption 3.2 do not hold, it is possible to use one-sided derivatives and still carry out IPA, as in [4]. Consequently, establishing the unbiasedness of $\partial Q_i(\boldsymbol{\theta})/\partial\theta_j$ and $\partial Q_i(\boldsymbol{\theta})/\partial\theta_j$ reduces to verifying the Lipschitz continuity of the sample functions $Q_i(\boldsymbol{\theta})$ and $L_i(\boldsymbol{\theta})$ with appropriate Lipschitz constants. To begin with, we will need the following additional assumptions.

Assumption 3.3. W.p. 1, all external processes are bounded, i.e., $0 < C_{\min} \le C(t) \le C_{\max} < \infty$, $r_i(t) \le R_i < \infty$, $i = 1, 2$, and feasible threshold parameters are also bounded from below, i.e., there exists some $\epsilon > 0$ such that $\theta_i \ge \epsilon > 0$, $i = 1, 2$.

Assumption 3.4. Let N_1 denote the number of exogenous events during interval $[0, T]$. Then, $E[N_1] < \infty$.

The proof of unbiasedness relies on two lemmas. First, Lemma 3.2 provides a bound for the expected number of events in the interval $[0, T]$. This is necessary because, as we have already seen, there may be potentially infinite ω-event chains in a sample path of this SFM.

Lemma 3.2. *Let N_T be the number of events occurring during interval $[0, T]$. Then, $E[N_T] < \infty$.*

Proof. See Appendix. □

Using Lemma 3.2, we can further bound all state and event time derivatives, as shown in the next lemma.

Lemma 3.3. *For all $t \in [0, T]$, $|x'_i(\theta, t)|$ is bounded w.p. 1, and for all $k = 1, 2, \ldots, N_T$, $|\tau'_k|$ is also bounded w.p. 1.*

Proof. See Appendix. □

Now, with the above two Lemmas, we are able to verify the Lipschitz continuity and finally establish the unbiasedness of IPA estimators.

Theorem 3.1. *Under Assumptions 3.1–3.3, the IPA estimators $\partial Q_i(\theta)/\partial\theta_j$ and $\partial L_i(\theta)/\partial\theta_j$, $i, j = 1, 2$, are unbiased estimates of $\mathrm{d}E[Q_i(\theta)]/\mathrm{d}\theta_j$ and $\mathrm{d}E[L_i(\theta)]/\mathrm{d}\theta_j$, respectively.*

Proof. See Appendix. □

3.5. User-centric and system-centric optimization

Recall that in the user-centric optimization approach, class i aims at minimizing the performance metric defined in (29), whereas in the system-centric approach the objective is to minimize a weighted sum of all classes' performance metrics as shown in (30). For the latter case, we use the IPA estimators (45) and (46) and the iterative scheme (31) with a constant time between iterations during which the system collects all observed data for evaluating $\frac{\partial Q_i(\boldsymbol{\theta})}{\partial\theta_j}$ and $\frac{\partial L_i(\boldsymbol{\theta})}{\partial\theta_j}$ and hence $\frac{\partial J(\boldsymbol{\theta})}{\partial\theta_j}$, $j = 1, 2$. This leads to a solution of the system-centric optimization problem under well-known appropriate technical conditions [15] on the step size sequence $\{\eta_n\}$.

In the case of user-centric optimization, however, each user has no information on the other's performance and no control over the other user's threshold. The resulting game is carried out by having users take turns, so that user i executes (31) with $\frac{\partial J_i(\boldsymbol{\theta})}{\partial\theta_i}$ evaluated after a time interval over which this user collects observed data for evaluating $\frac{\partial Q_i(\boldsymbol{\theta})}{\partial\theta_i}$ and $\frac{\partial L_i(\boldsymbol{\theta})}{\partial\theta_i}$ and hence $\frac{\partial J_i(\boldsymbol{\theta})}{\partial\theta_i}$. At the $(k+1)$th iteration, suppose it is class 1's turn to make a move: it executes (31) using the result of the previous step, denoted by $(\theta_{1,k}, \theta_{2,k})$, to evaluate $\theta_{1,k+1}$. Since the controllable thresholds are subject to (26), it follows that $\theta_{2,k+1} = \Theta - \theta_{2,k}$. It is easy to see that such a process does not generally converge. For example, when Θ is small and cannot satisfy the individual needs of either class, the game consists of oscillations between $(0, \Theta)$ and $(\Theta, 0)$. This is in contrast to the game considered in our earlier work [11] where the constraint (26) is not present and the user-centric optimization process does converge to a point $\boldsymbol{\theta}^u = (\theta_1^u, \theta_2^u)$. This, however, differs from the system-centric optimal $\boldsymbol{\theta}^s = (\theta_1^s, \theta_2^s)$. In fact, $J_i(\boldsymbol{\theta}^s) < J_i(\boldsymbol{\theta}^u)$ for both $i = 1, 2$ and the gap $J_i(\boldsymbol{\theta}^s) - J_i(\boldsymbol{\theta}^u) < 0$ reflects the inefficiency of "selfish play", sometimes also referred to as the "price of anarchy".

In order to enforce convergence in the resource contention game where (26) applies, we introduce a "negotiation scheme" as follows. At the kth iteration, suppose the current resource allocation is $(\theta_{1,k}, \theta_{2,k})$. Class 1 evaluates its next "candidate" control, denoted by $\theta^1_{1,k}$ through (31) using the IPA estimate of $\frac{\partial J_1(\boldsymbol{\theta})}{\partial \theta_1}$ from (45) and (46), and hence the control for class 2 as well, denoted by $\theta^1_{2,k} = \Theta - \theta^1_{1,k}$. Similarly, class 2 evaluates $\theta^2_{1,k}$ and $\theta^2_{2,k}$. Then, the new allocation for the next iteration is obtained from

$$\theta_{1,k+1} = \sum_{i=1}^{2} \zeta_i \theta^i_{1,k}, \qquad \theta_{2,k+1} = \sum_{i=1}^{2} \zeta_i \theta^i_{2,k} \tag{47}$$

where $\sum_{i=1}^2 \zeta_i = 1$ and ζ_i is the negotiation weight for class i (agreed upon in advance); this represents the class's relative "power" or, alternatively, a "price" that class i is willing to pay to influence the ultimate allocation. It is obvious that as ζ_1 varies from 0 to 1, the user-centric optimization result will range from the $J_1(\boldsymbol{\theta})$ optimum to the $J_2(\boldsymbol{\theta})$ optimum, and the following result gives a simple sufficient condition under which the user-centric optimization will also converge to the system-centric optimal point $\boldsymbol{\theta}^s$. However, it should be pointed out that this condition is quite restrictive. Generally, even when the negotiation scheme is used, there is still no guarantee that user-centric optimization will converge to $\boldsymbol{\theta}^s$. We also point out that this optimization scheme imposes synchronized updates of the competing users' gradient estimates and controls.

Theorem 3.2. *Let $\{\eta_{i,k}\}$, $k = 1, 2, \ldots$, be the step size sequence used by class $i = 1, 2$. If $\zeta_1\eta_{1,k} = \zeta_2\eta_{2,k}$, for all $k = 1, 2, \ldots$, then, if the system-centric optimization process converges to $\boldsymbol{\theta}^s$, the user centric optimization processes converge to $(\theta^u_1, \theta^u_2) = \boldsymbol{\theta}^u = \boldsymbol{\theta}^s$.*

Proof. See Appendix. □

This discussion has been based on the premise that sample paths of the SFM are available to the system optimizer and to the individual users. While this is possible in a simulated environment, in actuality the resource contention game takes place in a DES and what is available to the system and to the players is a DES sample path over $[0, T]$ denoted by $\xi_T^{\mathrm{DES}}(\boldsymbol{\theta})$. The system-centric objective function is denoted by $J_T^{\mathrm{DES}}(\boldsymbol{\theta})$ and the user-centric objective functions by $J_{i,T}^{\mathrm{DES}}(\boldsymbol{\theta})$, $i = 1, 2$. Therefore, we rewrite the iterative scheme (31) as

$$\theta_{i,k+1} = \theta_{i,k} - \eta_k H_{i,k}(\boldsymbol{\theta}_k; x(0), \xi_T^{\mathrm{DES}}), \quad k = 0, 1, \ldots$$

to emphasize the fact that the sample path from which data are obtainable is the one actually observed that involves discrete resource requests. Clearly, $H_{i,k}(\boldsymbol{\theta}_k; x(0), \xi_T^{\mathrm{DES}})$ is an estimate of $\mathrm{d}J_T^{\mathrm{DES}}(\boldsymbol{\theta})/\mathrm{d}\theta_i$ when the control is set at $\boldsymbol{\theta}_k$ for the system-centric optimization and of $\mathrm{d}J_{i,T}^{\mathrm{DES}}(\boldsymbol{\theta})/\mathrm{d}\theta_i$, $i = 1, 2$, for the user-centric optimization. None of these derivative estimates is in fact available, since the IPA estimators (45) and (46) are based on the SFM, an abstraction of the underlying DES. Therefore, (45) and (46) provide approximations of the performance derivative estimates obtained from the expressions derived through the SFM, but using data available from the actual DES. What facilitates this process is the fact that (45) and (46) are evaluated on an event by event basis; thus, we need to identify each SFM event with a DES event, as explained next.

An exogenous SFM event corresponds to a jump in the process $\{C(t)\}$ or $\{r_i(t)\}$, $i = 1, 2$. In the DES, a simple rate estimator is used to measure $r_i(t)$, periodically updated over t; if $|r_i(t) - r_i(t - \Delta)| > \epsilon$ for some adjustable Δ, ϵ, then we identify this as an exogenous SFM event. From (39), however, there is no effect from such events on the actual estimators (45) and (46). The same applies to $|C(t) - C(t - \Delta)| > \epsilon$.

For any endogenous event "$x_i(t)$ becomes 0 ", observe that there is no difference between DES and SFM: when this event occurs in a DES sample path, we identify it with a SFM event that empties a queue. We then simply apply (42).

On the other hand, for an endogenous event "$x_i(t)$ becomes θ_i" in the SFM, we have to recognize that in a DES the queue content may "chatter" near θ_i. For example, suppose at time t, the class i queue becomes full in the DES, i.e., $x_i(t) = \lceil\theta_i\rceil$ (where $\lceil\theta_i\rceil$ is the ceiling function). If $r_i(t) \geq c_i(t)$, then it follows from (20) that the SFM enters a FP for class i. However, in the DES sample path it is still possible that a "resource service completion" event takes place next at time $t_1 > t$, resulting in $x_i(t_1) = \lceil\theta_i\rceil - 1 < \theta_i$, whereas in the SFM we would have $x_i(t_1) = \theta_i$. Moreover, suppose that at $t_2 > t_1$, a "request arrival" event occurs in the DES, therefore $x_i(t_2) = \lceil\theta_i\rceil$ once again. Since in the SFM we still have $x_i(t_1) = \theta_i$, this event should be ignored. In other words, during the interval $[t, t_2]$, the queue length of the actual DES system "chatters" between $\lceil\theta_i\rceil$ and $\lceil\theta_i\rceil - 1$, while in the SFM a FP is taking place. This raises the issue of properly identifying the start and end of a FP in the SFM. We resolve this by calculating c_i through (23) and (22) and measurements of $r_i(t)$ and $C(t)$ from the DES using simple rate estimators as mentioned above. Then, we can identify an event at τ_k as initiating a FP for class i if the following condition holds:

$$x_i(\tau_{k-1}) < \theta_i, \quad x_i(\tau_k) = \theta_i, \qquad c_i(\tau_k) \leq r_i(\tau_k)$$

at which point we can use (40) and (41). Similarly, we detect the end of a FP at τ_k if

$$x_i(\tau_{k-1}) = \theta_i, \quad x_i(\tau_k) < \theta_i, \qquad c_i(\tau_k) > r_i(\tau_k).$$

Finally, ω-events are easy to detect since they involve a simple timer initiated at the associated inflow change event which is also easy to detect once its type (exogenous, endogenous, or another ω-event) is identified.

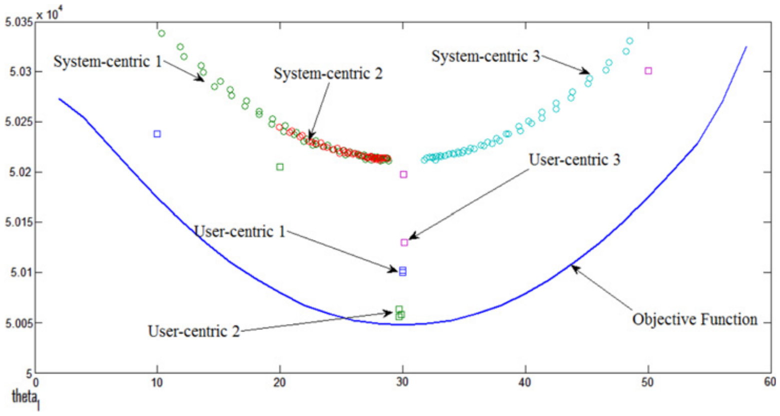

Fig. 4. Simulation example: Symmetric case.

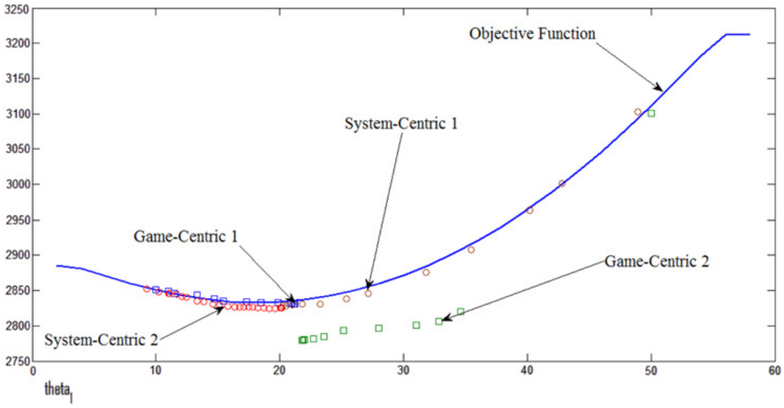

Fig. 5. Simulation example: Asymmetric case.

4. Resource contention game simulation examples

We present numerical results obtained from simulating a two-class FCFS queueing system subject to (26) with $\Theta = 60$. In this system, requests of both classes arrive according to Markov Modulated Poisson Processes (MMPP), with mean interarrival times uniformly distributed over [0.6 s, 1.2 s] and [0.6 s, 1 s] respectively. The overall service capacity $C(t)$ is a piecewise constant random process with mean 0.9 s and times between changes in the capacity value are exponentially distributed with mean 500 s. In what follows, we will use the same negotiation weights and step sizes for both classes when applying user-centric optimization, therefore, based on Theorem 3.2, we expect the user-centric and system-centric optima to coincide.

Fig. 4 shows simulation results of both system-centric and user-centric optimization (three different sample paths shown for each). In this case, the weights in the system-centric objective (30) are set to $w_1 = w_2 = 1$. In the user-centric optimization, the two classes have the same relative weights for workload and loss rate, i.e. $\frac{\gamma_{1,1}}{\gamma_{2,1}} = \frac{\gamma_{1,2}}{\gamma_{2,2}}$ in (29). The "actual" system-centric objective function, which was obtained by exhaustive simulation of the underlying DES averaged over 50 sample paths, is symmetric with an actual system-centric optimal point (30, 30) as seen in the figure.

Fig. 5 shows another example for the same system (two different sample paths shown), except that classes now have different relative weights, $\gamma_{1,1} = 0.8$, $\gamma_{2,1} = 2500$ for class 1 and $\gamma_{1,2} = 0.6$, $\gamma_{2,2} = 2750$ for class 2, so that $\frac{\gamma_{1,1}}{\gamma_{2,1}} > \frac{\gamma_{1,2}}{\gamma_{2,2}}$ (class 1 puts more emphasis on reducing workload) and the objective function is asymmetric with a system-centric optimal point (20, 40), shifting to a smaller θ_1.

In both cases, the user-centric and system-centric optimization processes converge to a point close to the "true" optimum (obtained by exhaustive brute-force simulation), illustrating the effectiveness of our method. Moreover, in contrast to the result in [11], under the sufficient condition given in Theorem 3.2, there is no gap between the corresponding optimal points.

5. Conclusions

We have presented a general resource contention game setting using SFMs, where multiple user classes with different class-dependent objectives compete for a sharable resource. In this setting, IPA estimators were obtained for the derivatives of various class-dependent objectives through which gradient-based optimization can be carried out, either system-centric for a system-wide objective or user-centric where each user class optimizes its own performance metric. We derived explicit solutions for a specific game in which the competing user classes employ threshold control policies and service is provided on a First Come First Serve (FCFS) basis. The unbiasedness of the IPA estimators was established in this case and it was shown that under certain conditions the system-centric and user-centric optimization solutions coincide. We also discussed how to use the SFM-based IPA estimators evaluated with data obtained from the actual observed DES sample paths and illustrated the effectiveness of our approach through simulation examples in which we contrasted system-centric and user-centric optimization.

Ongoing research is directed at modifying the iterative scheme followed by the game so that it becomes asynchronous. In other words, users may be allowed to take control actions (changing their respective threshold parameters) whenever they feel their IPA estimates are sufficiently accurate rather than the synchronized fashion we have adopted thus far. Moreover, we are exploring different games where system-centric and user-centric optima can be guaranteed to coincide.

Acknowledgements

The second author was supported in part by the National Science Foundation under Grant EFRI-0735794, by AFOSR under grants FA9550-07-1-0361 and FA9550-09-1-0095, by DOE under grant DE-FG52-06NA27490, and by ONR under grant N00014-09-1-1051.

Appendix

Proof of Lemma 3.2. Let N_1, N_2 and N_3 denote the number of exogenous events, endogenous events and ω-events respectively. First, by Assumption 3.4,

$$E[N_1] \leq n_1 \tag{48}$$

for some positive number n_1.

For endogenous events, they can be further classified into (i) events that initiate FPs, which we will call "ρ events", and (ii) events that initiate EPs, which we will call "σ events". Let N_ρ and N_σ denote the number of ρ events and σ events respectively, so that $N_2 = N_\rho + N_\sigma$. By definition, each ρ event corresponds to one FP, hence, the number of ρ events is the same as the number of FPs. For any FP, based on the system dynamics (20), it is either ended by an exogenous event that increases $C(t)$ or decreases $r_i(t)$ or by an ω-event that increases $c_i(t)$ through (23). The number of FPs that are ended by exogenous events is obviously bounded by the total number of exogenous events N_1. Next, consider a FP that is ended by an ω-event and let τ be its starting time. There are two cases regarding the ω-event that ends the FP.

Case 1: The ω-event that ends the FP is induced by an inflow change event before τ. In this case, because the system is in a NBP before τ, i.e., $\alpha_i(t) = r_i(t)$, the inflow change event must be an exogenous event that increases r_i. Therefore, the number of FPs in this case is bounded by the total number of exogenous events N_1.

Case 2: The ω-event that ends the FP is induced by an inflow change event at or after τ. Suppose this inflow change event occurs at $\tau_p \geq \tau$. Then, based on Lemma 1 in [11], $\tau_p + \omega(\tau_p) \geq \tau + \omega(\tau)$ and the duration of the FP is $\tau_p + \omega(\tau_p) - \tau \geq \omega(\tau)$. By the definition of $\omega(t)$ in (22), since $x_i(\tau) = \theta_i$, we have $\omega(\tau) = \frac{\theta_i}{c_i(\tau)}$. By Assumption 3.2, $\theta_i \geq \epsilon > 0$, and $c_i(\tau) < C(\tau) \leq C_{\max}$, therefore, the length of any FP is such that

$$\tau_p + \omega(\tau_p) - \tau \geq \frac{\epsilon}{C_{\max}}$$

which implies that the number of such FPs over an interval $[0, T]$ is bounded by $T/(\epsilon/C_{\max}) = TC_{\max}/\epsilon$.

Combing the two cases above, the number of FPs that are ended by ω-events is bounded by $N_1 + \frac{TC_{\max}}{\epsilon}$. Recalling that the remaining FPs are ended by exogenous events and that the number of FPs is given by N_ρ, we have

$$N_\rho \leq 2N_1 + \frac{TC_{\max}}{\epsilon}. \tag{49}$$

Similarly, the number of σ events, N_σ, is equal to the number of EPs in $[0, T]$. Since EPs can only be ended by exogenous events, the number of EPs is bounded by the total number of exogenous events, i.e.,

$$N_\sigma \leq N_1. \tag{50}$$

Combining (49), (50) and (48), we have

$$\begin{aligned} E[N_2] &\leq 3E[N_1] + \frac{TC_{\max}}{\epsilon} \\ &\leq 3n_1 + \frac{TC_{\max}}{\epsilon} \equiv n_2. \end{aligned} \tag{51}$$

Therefore, the expected number of endogenous events is also bounded.

Finally, for ω-events, as shown in Section 3.1, they can be induced in three ways:

(i) Induced by an exogenous event. Obviously the number of such ω-events is bounded by N_1.

(ii) Induced by an endogenous event. Obviously the number of such ω-events is bounded by N_2.

(iii) Induced by another ω-event which takes place while $x_i(t) = \theta_i$, $i = 1, 2$. For ω-events in this category, as discussed in Section 3.1, they will form multiple sequences of ω-events, referred to as "ω-event chains". All such chains are initiated by either an exogenous or an endogenous event, and exist only in FPs. Within any one of these chains, from the system dynamics (22), the time between any ω-event at τ' and the one that immediately precedes it is, by the definition of $\omega(t)$ in (22), is given by $\frac{\theta_i}{c_i(\tau'^-)}$. Since $\theta_i \geq \epsilon > 0$ and $c_i(\tau) < C(\tau) \leq C_{\max}$ by Assumption 3.2, we have $\frac{\theta_i}{c_i(\tau'^-)} \geq \frac{\epsilon}{C_{\max}}$, i.e., the time between two consecutive events in the chain is bounded by $\frac{\epsilon}{C_{\max}}$. On the other hand, since the length of any chain is bounded by T, the number of ω-events in any one chain is bounded by $T/(\epsilon/C_{\max}) = TC_{\max}/\epsilon$. In addition, since all chains are initiated by either an exogenous or an endogenous event, it follows that the number of ω-event chains is bounded by $N_1 + N_2$. We conclude from this analysis that the number of ω-events in this category is bounded by $(N_1 + N_2) \cdot \frac{TC_{\max}}{\epsilon}$.

Combining all types of ω-events above, we have

$$N_3 \leq (N_1 + N_2) \cdot \left(\frac{TC_{\max}}{\epsilon} + 1 \right).$$

Together with (48) and (51), this implies

$$\begin{aligned} E[N_3] &= E\left[(N_1 + N_2) \cdot \left(\frac{TC_{\max}}{\epsilon} + 1 \right) \right] \\ &\leq (E[N_1] + E[N_2]) \cdot \left(\frac{TC_{\max}}{\epsilon} + 1 \right) \\ &\leq (n_1 + n_2) \cdot \left(\frac{TC_{\max}}{\epsilon} + 1 \right) \equiv n_3. \end{aligned}$$

Thus,

$$\begin{aligned} E[N_T] &= E[N_1] + E[N_2] + E[n_3] \\ &\leq n_1 + n_2 + n_3 < \infty \end{aligned}$$

and the result of the Lemma is established. □

Proof of Lemma 3.3. First, using Lemma 3.2, we know that $E[N_T]$ is bounded, i.e., N_T is finite w.p. 1. Since, from (38), we know that $x'_{i,j}(t)$ is fixed between any two consecutive events, we only have to prove the boundedness of $\{x'_{i,j}(\tau_k^+)\}$ and $\{\tau'_{kj}\}$ for all $k \in \{0, 1, 2, \ldots, N_T\}$. We proceed to prove this by induction on k. When $k = 0$, $x'_{i,j}(\tau_0^+) = 0$, $i = 1, 2$, and $\tau'_{0j} = 0$, which are obviously bounded. Assume that for some integer n, $0 \leq n < N_T$, and for all $k \leq n$, $|x'_{i,j}(\tau_k^+)|$ and $|\tau'_{kj}|$ are bounded. At τ_{n+1}, there are four possible event types that can occur:

Case 1: An exogenous event occurs at τ_{n+1}. Then, from (39), we know that

$$\begin{aligned} &\tau'_{n+1,j} = 0 \\ &x'_{i,j}(\tau_{n+1}^+) = x'_{i,j}(\tau_{n+1}^-) = x'_{i,j}(\tau_n^+) \quad i, j = 1, 2. \end{aligned}$$

Since, by the induction hypothesis, $|x'_{i,j}(\tau_n^+)|$ is bounded, it follows that $|x'_{i,j}(\tau_{n+1}^+)|$ and $\tau'_{n+1,j}$ are also bounded.

Case 2: A ρ event (defined in the proof of Lemma 3.2) occurs at τ_{n+1}. From (40) to (41), if $j = i$,

$$\begin{aligned} &\tau'_{n+1,j} = \frac{1 - x'_{ij}(\tau_{n+1}^-)}{r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-)} = \frac{1 - x'_{ij}(\tau_n^+)}{r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-)} \\ &x'_{ij}(\tau_{n+1}^+) = 1 \quad i, j = 1, 2 \end{aligned}$$

where $x'_{ij}(\tau_{n+1}^+)$ is obviously bounded. Regarding $\tau'_{n+1,j}$,

$$\left| \tau'_{n+1,j} \right| \leq \frac{1 + \left| x'_{ij}(\tau_n^+) \right|}{\left| r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-) \right|}$$

where $|x'_{ij}(\tau_n^+)|$ is bounded by the induction hypothesis, and $|r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-)| > 0$, otherwise (i.e., if $|r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-)| = 0$), we cannot have $x_i(\tau_{n+1}^-) < \theta_i$ and $x_i(\tau_{n+1}^+) = \theta_i$. Therefore, there exists a strictly positive number $\varkappa$, such that $|r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-)| > \varkappa$ and we get

$$\left| \tau'_{n+1j} \right| \leq \frac{1 + \left| x'_{ij}(\tau_n^+) \right|}{\left| r_i(\tau_{n+1}^-) - c_i(\tau_{n+1}^-) \right|} \leq \frac{1 + \left| x'_{ij}(\tau_n^+) \right|}{\varkappa}$$

which implies that $|\tau'_{n+1,j}|$ is also bounded.

On the other hand, if $j \neq i$, then

$$\tau'_{n+1,j} = \frac{-x'_{ij}(\tau^-_{n+1})}{r_i(\tau^-_{n+1}) - c_i(\tau^-_{n+1})}$$
$$x'_{i,j}(\tau^+_{n+1}) = x'_{i,j}(\tau^-_{n+1}) = x'_{i,j}(\tau^+_n)$$

where $x'_{i,j}(\tau^+_{n+1})$ is obviously bounded by the induction hypothesis. Regarding $\tau'_{n+1,j}$, with a similar argument we have

$$\left|\tau'_{n+1,j}\right| \leq \frac{\left|x'_{ij}(\tau^+_n)\right|}{\left|r_i(\tau^-_{n+1}) - c_i(\tau^-_{n+1})\right|}$$

where $|r_i(\tau^-_{n+1}) - c_i(\tau^-_{n+1})|$ is lower bounded, and together with the induction hypothesis, we can prove the boundedness of $|\tau'_{n+1,j}|$ once again.

We conclude from the above analysis, that $|x'_{i,j}(\tau^+_{n+1})|$ and $\tau'_{n+1,j}$ are bounded in Case 2.

Case 3: A σ event (defined in the proof of Lemma 3.2) occurs at τ_{n+1}. Then, according to (42)

$$\tau'_{n+1,j} = \frac{-x'_{ij}(\tau^-_{n+1})}{r_i(\tau^-_{n+1}) - c_i(\tau^-_{n+1})}$$
$$x'_{ij}(\tau^+_{n+1}) = 0 \quad i = 1, 2.$$

Obviously $|x'_{ij}(\tau^+_{n+1})|$ is bounded, and using a similar argument as in Case 2, it is easy to prove the boundedness of $|\tau'_{n+1,j}|$.

Case 4: An ω-event occurs at τ_{n+1}. Suppose that the inflow change event that induces this ω-event occurs at time τ_m, where $\tau_m \in \{\tau_0, \ldots, \tau_n\}$, and $\tau_m + \omega(\tau_m) = \tau_{n+1}$. Based on (44), (43) and (37), we have

$$\tau'_{n+1,j} = \frac{\sum_{i=1}^{2} x'_{i,j}(\tau^-_m) + \left[\sum_{i=1}^{2} f_{i,m}(\tau^-_m) + C(\tau^+_m)\right]\tau'_{m,j}}{C(\tau^+_n)}$$
$$x'_{ij}(\tau^+_{n+1}) = x'_{ij}(\tau^+_n) + \left[\alpha_i(\tau^-_{n+1}) - \alpha_i(\tau^+_{n+1}) + c_i(\tau^+_{n+1}) - c_i(\tau^-_{n+1})\right] \cdot \tau'_{n+1,j}.$$

By (17) and Assumption 3.3,

$$\begin{aligned}\sum_{i=1}^{2} f_{i,m}(\tau^-_m) &= \sum_{i=1}^{2} \alpha_{i,m}(\tau^-_m) - C(\tau^-_m)\\ &\leq r_1(\tau^-_m) + r_2(\tau^-_m) + C(\tau^-_m)\\ &\leq R_1 + R_2 + C_{\max}.\end{aligned}$$

Also by Assumption 3.3, $C(\tau^+_n) \geq C_{\min}$, therefore,

$$\left|\tau'_{n+1,j}\right| \leq \frac{\sum_{i=1}^{2}\left|x'_{i,j}(\tau^-_m)\right| + (R_1 + R_2 + 2C_{\max}) \cdot \left|\tau'_{m,j}\right|}{C_{\min}}. \tag{52}$$

Similarly,

$$\begin{aligned}\left|\alpha_i(\tau^+_n) - c_i(\tau^+_n) - \alpha_i(\tau^+_{n+1}) + c_i(\tau^+_{n+1})\right| &\leq \left|\alpha_i(\tau^+_n)\right| + \left|c_i(\tau^+_n)\right| + \left|\alpha_i(\tau^+_{n+1})\right| + \left|c_i(\tau^+_{n+1})\right|\\ &\leq R_i + C_{\max} + R_i + C_{\max}\end{aligned}$$

and we get:

$$\left|x'_{ij}(\tau^+_{n+1})\right| \leq \left|x'_{ij}(\tau^+_n)\right| + 2 \cdot (R_i + C) \cdot \left|\tau'_{n+1,j}\right|. \tag{53}$$

Since $\tau_m \in \{\tau_0, \ldots, \tau_n\}$, the induction hypothesis applies to $|x'_{i,j}(\tau^-_m)|$ and $|\tau'_{m,j}|$ in (52), which implies the boundedness of $|\tau'_{n+1,j}|$. Then, $|x'_{ij}(\tau^+_{n+1})|$ is also bounded from (53).

This concludes the inductive step and completes the proof of the lemma. □

Proof of Theorem 3.1. Invoking Lemma A2 in [16], the unbiasedness of the IPA estimators $\frac{\partial L_i(\boldsymbol{\theta})}{\partial \theta_j}$ and $\frac{\partial Q_i(\boldsymbol{\theta})}{\partial \theta_j}$ relies on the Lipschitz continuity of $L_i(\boldsymbol{\theta})$ and $Q_i(\boldsymbol{\theta})$. From (46), and together with Assumption 3.3, we have

$$\left|\frac{\partial L_i(\boldsymbol{\theta})}{\partial \theta_j}\right| = \left|\frac{1}{T}\sum_{k \in \Psi_i}\left[\left[r_i\left(\tau^+_{k-1}\right) - \alpha_i\left(\tau^+_{k-1}\right)\right]\left(\tau'_{k,j} - \tau'_{k-1,j}\right)\right]\right|$$

$$\begin{aligned}
&\le \frac{1}{T}\sum_{k\in\Psi_i}\left[[R_i + R_i]\left(\left|\tau'_{k,j}\right| + \left|\tau'_{k-1,j}\right|\right)\right]\\
&\le \frac{1}{T}\cdot N_T \cdot \left[2R_i\left(\left|\tau'_{k,j}\right| + \left|\tau'_{k-1,j}\right|\right)\right].
\end{aligned}$$

Using Lemmas 3.2 and 3.3, it follows that

$$\left|\frac{\partial L_i(\boldsymbol{\theta})}{\partial\theta_j}\right| < B_1 < \infty$$

for some finite B_1 with $\frac{1}{T}\cdot N_T\cdot[2R_i(|\tau'_{k,j}| + |\tau'_{k-1,j}|)] < B_1$. Therefore,

$$\left|\Delta L_i(\boldsymbol{\theta})\right| = \left|\frac{\partial L_i(\boldsymbol{\theta})}{\partial\theta_j}\right|\left|\Delta\theta_j\right| < B_1\left|\Delta\theta_j\right|. \tag{54}$$

In other words, $L_i(\boldsymbol{\theta})$ is Lipschitz Continuous with finite Lipschitz constant B_1.

For $Q_i(\theta)$, by definition

$$Q_i(\boldsymbol{\theta}) = \frac{1}{T}\int_0^T x_i(t)\mathrm{d}t, \quad i = 1, 2$$

hence,

$$\left|\frac{\partial Q_i(\boldsymbol{\theta})}{\partial\theta_j}\right| = \frac{1}{T}\left|\int_0^T x'_{ij}(t)\mathrm{d}t\right| \le \frac{1}{T}\int_0^T \left|x'_{ij}(t)\right|\mathrm{d}t.$$

Based on Lemma 3.3, we know that $|x'_{ij}(t)|$ is bounded, therefore, the above equation guarantees the boundedness of $|\partial Q_i(\theta)/\partial\theta_j|$. i.e.,

$$\left|\frac{\partial Q_i(\boldsymbol{\theta})}{\partial\theta_j}\right| < B_2 < \infty$$

where B_2 is such that $\frac{1}{T}\int_0^T |x'_{ij}(t)|\mathrm{d}t < B_2$, and we get

$$\left|\Delta Q_i(\boldsymbol{\theta})\right| = \left|\frac{\partial Q_i(\boldsymbol{\theta})}{\partial\theta_j}\right|\left|\Delta\theta_j\right| < B_2\left|\Delta\theta_j\right|. \tag{55}$$

Thus, $Q_i(\boldsymbol{\theta})$ is also Lipschitz continuous with finite Lipschitz constant B_2. □

Proof of Theorem 3.2. Based on (47), for any iteration k,

$$\begin{aligned}
\theta_{1,k+1} &= \zeta_1\theta^1_{1,k} + \zeta_2\theta^2_{1,k}\\
&= \zeta_1\left(\theta_{1,k} - \eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1}\right) + \zeta_2(\Theta - \theta^2_{2,k})\\
&= \zeta_1\left(\theta_{1,k} - \eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1}\right) + \zeta_2\left(\Theta - \left(\theta_{2,k} - \eta_{2,k}\cdot\frac{\partial J_2(\theta)}{\partial\theta_2}\right)\right)\\
&= \zeta_1\left(\theta_{1,k} - \eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1}\right) + \zeta_2\left(\theta_{1,k} + \eta_{2,k}\cdot\frac{\partial J_2(\theta)}{\partial\theta_2}\right)\\
&= (\zeta_1 + \zeta_2)\cdot\theta_{1,k} - \zeta_1\eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1} + \zeta_2\eta_{2,k}\cdot\frac{\partial J_2(\theta)}{\partial\theta_2}\\
&= \theta_{1,k} - \zeta_1\eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1} + \zeta_2\eta_{2,k}\cdot\frac{\partial J_2(\theta)}{\partial(\Theta - \theta_1)}\\
&= \theta_{1,k} - \zeta_1\eta_{1,k}\cdot\frac{\partial J_1(\theta)}{\partial\theta_1} - \zeta_2\eta_{2,k}\cdot\frac{\partial J_2(\theta)}{\partial\theta_1}.
\end{aligned}$$

Then if $\zeta_1\eta_{1,k} = \zeta_2\eta_{2,k}$, which we denote as η_k, the above equation is further reduced to

$$\theta_{1,k+1} = \theta_{1,k} - \eta_k\cdot\frac{\partial J_1(\theta)}{\partial\theta_1} - \eta_k\cdot\frac{\partial J_2(\theta)}{\partial\theta_1}$$

$$
\begin{aligned}
&= \theta_{1,k} - \eta_k \left(\frac{\partial J_1(\theta)}{\partial \theta_1} + \frac{\partial J_2(\theta)}{\partial \theta_1} \right) \\
&= \theta_{1,k} - \eta_k \cdot \frac{\partial J(\theta)}{\partial \theta_1}. \qquad (56)
\end{aligned}
$$

Similarly,

$$\theta_{2,k+1} = \theta_{2,k} - \eta_k \cdot \frac{\partial J(\theta)}{\partial \theta_2}. \qquad (57)$$

From (56) and (57), one can see that under the given condition the user-centric optimization is equivalent to a system-centric optimization using a step size sequence $\{\eta_k\}$, therefore both will converge to the same point. □

References

[1] C.G. Cassandras, J. Lygeros (Eds.), Stochastic Hybrid Systems, Taylor and Francis, 2006.

[2] D. Anick, D. Mitra, M.M. Sondhi, Stochastic theory of a data-handling system with multiple sources, Bell Syst. Tech. J. 61 (1982) 1871–1894.

[3] B. Liu, Y. Guo, J. Kurose, D. Towsley, W.B. Gong, Fluid simulation of large scale networks: issues and tradeoffs, in: Proc. of the Intl. Conf. on Parallel and Distributed Processing Techniques and Applications, June 1999, pp. 2136–2142.

[4] C.G. Cassandras, Y. Wardi, B. Melamed, G. Sun, C.G. Panayiotou, Perturbation analysis for on-line control and optimization of stochastic fluid models, IEEE Trans. Automat. Control AC-47 (8) (2002) 1234–1248.

[5] C.G. Cassandras, S. Lafortune, Introduction to Discrete Event Systems, 2nd ed., Springer, 2008.

[6] C.G. Cassandras, Stochastic flow systems: modeling and sensitivity analysis, in: C.G. Cassandras, J. Lygeros (Eds.), Stochastic Hybrid Systems, Taylor and Francis, 2006, pp. 139–167.

[7] Y. Wardi, R. Adams, B. Melamed, A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case, IEEE Trans. Automat. Control 55 (1) (2010) 89–103.

[8] C. Yao, Christos.G. Cassandras, Using infinitesimal perturbation analysis of stochastic flow models to recover performance sensitivities of single-server queueing systems, Technical Report, Center of Information and Systems Engineering, Boston University, 2010. http://vita.bu.edu/cyao/SFMEss.pdf.

[9] G. Sun, C.G. Cassandras, C.G. Panayiotou, Perturbation analysis of multiclass stochastic fluid models, J. Discrete Event Dyn. Syst. 14 (2004) 267–307.

[10] M. Chen, J.-Q. Hu, M.C. Fu, Perturbation analysis of a dynamic priority call center, IEEE Trans. Automat. Control (2010) (in press).

[11] C. Yao, C.G. Cassandras, Perturbation analysis and optimization of multiclass multiobjective stochastic flow models, in: Proc. of 48th IEEE Conference of Decision and Control, 2009, pp. 914–919.

[12] C.G. Cassandras, Y. Wardi, C.G. Panayiotou, C. Yao, Perturbation analysis and optimization of stochastic hybrid systems, Eur. J. Control (2010) (in press).

[13] C.G. Cassandras, G. Sun, C.G. Panayiotou, Y. Wardi, Perturbation analysis and control of two-class stochastic fluid models for communication networks, IEEE Trans. Automat. Control 48 (5) (2003) 770–782.

[14] C. Yao, C.G. Cassandras, Perturbation analysis and optimization of multiclass multiobjective stochastic flow models, Technical Report, Div. of Systems Engineering, Boston University, 2009. http://people.bu.edu/cyao/files/techreport.pdf.

[15] H.J. Kushner, G.G. Yin, Stochastic Approximation Algorithms and Applications, Springer-Verlag, 1997.

[16] R.Y. Rubinstein, A. Shapiro, Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method, John Wiley and Sons, New York, New York, 1993.