Front. Electr. Electron. Eng. China 2011, 6(3): 453–467
DOI 10.1007/s11460-011-0166-7

RESEARCH ARTICLE

Chen YAO, Christos G. CASSANDRAS

Perturbation analysis of stochastic hybrid systems and applications to resource contention games

Abstract We provide an overview of the recently developed general infinitesimal perturbation analysis (IPA) framework for stochastic hybrid systems (SHSs), and establish some conditions under which this framework can be used to obtain unbiased performance gradient estimates in a particularly simple and efficient manner. We also propose a general scheme for systematically deriving an abstraction of a discrete event system (DES) in the form of an SHS. Then, as an application of the general IPA framework, we study a class of stochastic non-cooperative games termed "resource contention games" modeled through stochastic flow models (SFMs), where two or more players (users) compete for the use of a sharable resource. Simulation results are provided for a simple version of such games to illustrate and contrast system-centric and user-centric optimization.

Keywords stochastic flow model (SFM), perturbation analysis, stochastic hybrid system (SHS), resource contention games, cyber-physical systems

1 Introduction

Arguably the most comprehensive modeling framework for the study of complex dynamic systems is that of *stochastic hybrid systems* (SHSs). Hybrid systems consist of interacting components, some with time-driven and others with event-driven dynamics. For example, electromechanical components governed by time-driven dynamics become a hybrid system when interacting through a communication network whose behavior is event-driven. Indeed, so-called "cyber-physical systems" are explicitly designed to allow discrete event components (e.g., embedded microprocessors, sensor networks) to be integrated with physical components (e.g., generators in a power grid, engine parts in an automotive vehicle). Given that they must also operate in the presence of uncertainty, an appropriate framework must include the means to represent stochastic effects, be they purely random or the result of adversarial disturbances.

The basis for a hybrid system modeling framework is often provided by a *hybrid automaton*. In a hybrid automaton, discrete events (either controlled or uncontrolled) cause transitions from one discrete state (or "mode") to another. While operating at a particular mode, the system's behavior is usually described by differential equations. 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 Ref. [1] along with various classes of SHS which exhibit different properties or suit different types of applications. The performance of an SHS is generally hard to estimate because of the absence of closed-form expressions capturing the dependence of interesting performance metrics on design or control parameters. Consequently, we lack the ability to systematically adjust such parameters for the purpose of improving — let alone optimizing — performance. In the domain of pure discrete event systems (DESs), it was discovered in the early 1980s that event-driven dynamics give rise to state trajectories (sample paths) from which one can very efficiently and nonintrusively extract sensitivities of various performance metrics with respect to at least certain types of design or control parameters. This has led to the development of a theory for perturbation analysis in DES [2–4], the most successful branch of which is *infinitesimal perturbation analysis* (IPA) due to its simplicity and ease of implementation. Using IPA, one obtains unbiased estimates of performance metric gradients that can be incorporated into standard

Received March 18, 2011; accepted May 10, 2011

Chen YAO, Christos G. CASSANDRAS (✉)
Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA
E-mail: cgc@bu.edu

gradient-based algorithms for optimization purposes. However, IPA estimates become biased (hence, unreliable for control purposes) when dealing with various aspects of DES that cause significant discontinuities in sample functions of interest. Such discontinuities normally arise when a parameter perturbation causes the order in which events occur to be affected and this event order change may violate a basic "commuting condition" [4]. When this happens, one must resort to significantly more complicated methods for deriving unbiased gradient estimates [2].

In recent years, it was shown that IPA can also be applied to at least some classes of SHS and yield simple unbiased gradient estimators that can be used for optimization purposes. In particular, *stochastic flow* (or *fluid*) *models* (SFMs), as introduced in Ref. [5], are a class of SHS where the time-driven component captures general-purpose flow dynamics and the event-driven component describes switches, controlled or uncontrolled, that alter the flow dynamics. What is attractive about SFMs is that they can be viewed as abstractions of complex stochastic DES which retain their essential features for the purpose of control and optimization. In fact, fluid models have a history of being used as abstractions of DES. Introduced in Ref. [6], fluid models have been shown to be very useful in simulating various kinds of high speed networks [7], manufacturing systems [8] and, more generally, settings where users compete over different sharable resources. It should be stressed that fluid models may not always provide accurate representations for the purpose of analyzing the performance of the underlying DES. What we are interested in, however, is control and optimization, in which case the value of a fluid model 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. While in most traditional fluid models the flow rates involved are treated as fixed parameters, an SFM has the extra feature of treating flow rates as *stochastic processes*. With only minor technical conditions imposed on the properties of such processes, the use of IPA has been shown to provide simple gradient estimators for stochastic resource contention systems that include blocking phenomena and a variety of feedback control mechanisms [9 – 12].

Until recently, the use of IPA was limited to a class of SFMs and was based on exploiting the special structure of specific systems. However, a major advance was brought about by the unified framework introduced in Ref. [10] and extended in Ref. [13] placing IPA in the general context of stochastic hybrid automata with arbitrary structure. As a result, IPA may now be applied to arbitrary SHS for the purpose of performance gradient estimation and, therefore, gradient-based optimization through standard methods.

In this paper, we begin with an overview of this general IPA framework for SHS. Our emphasis is on the main concepts and key results that will enable the reader to apply the general IPA methodology through a set of simple equations driven by observable system data; additional details on this material can be found in Ref. [13]. We then establish properties of the resulting gradient estimators (first reported in Ref. [14] without proof) that justify its applicability even in the absence of detailed models for the time-driven components in some cases. Next, we focus on DES and address the issue of obtaining an SHS model from an arbitrary DES as a means of abstraction that facilitates its analysis for control and optimization purposes. We specifically introduce a new scheme for systematically performing this abstraction process. An application of the general IPA framework in Ref. [13] is enabling the use of IPA in multi-agent multi-objective settings that include resource contention games, i.e., situations where two or more "players" compete for a shared resource. Thus, in the last part of the paper we analyze a class of resource contention games first studied in Ref. [14], i.e., situations where two or more "players" compete for a shared resource.

The paper is organized as follows. In Sect. 2 we present the general IPA framework for SHS. We then identify in Sect. 3 properties which provide sufficient conditions under which IPA is particularly simple and efficient, requiring no detailed information on the probabilistic characterizations of the random processes involved and minimal or no knowledge of the time-driven dynamics of the SHS within different discrete states. In Sect. 4 we propose a general scheme to abstract a DES to a stochastic hybrid automaton. In Sect. 5 we study resource contention games abstracted as SHS from both the point of view of system-centric performance and the user-centric approach where each user optimizes its own performance metric. For illustrative purposes, we include a simple example with numerical results contrasting these two optimization viewpoints.

2 General framework for perturbation analysis of stochastic hybrid systems

We begin by adopting a standard hybrid automaton formalism to model the operation of an SHS [1]. Thus, let $q \in Q$ (a countable set) denote the discrete state (or mode) and $x \in X \subseteq \mathbb{R}^n$ denote the continuous state. Let $\upsilon \in \Upsilon$ (a countable set) denote a discrete control input and $u \in U \subseteq \mathbb{R}^m$ a continuous control input. Similarly, let $\delta \in \Delta$ (a countable set) denote a discrete disturbance input and $d \in D \subseteq \mathbb{R}^p$ a continuous disturbance input. The state evolution is determined by means of 1) a vector field $f : Q \times X \times U \times D \to X$, 2) an invariant

(or domain) set Inv: $Q \times \Upsilon \times \Delta \to 2^X$, 3) a guard set Guard: $Q \times Q \times \Upsilon \times \Delta \to 2^X$, and 4) a reset function $r : Q \times Q \times X \times \Upsilon \times \Delta \to X$.

A sample path of such a system consists of a sequence of intervals of continuous evolution followed by a discrete transition. The system remains at a discrete state q as long as the continuous (time-driven) state x does not leave the set $\text{Inv}(q, \upsilon, \delta)$. If x reaches a set $\text{Guard}(q, q', \upsilon, \delta)$ for some $q' \in Q$, a discrete transition can take place. If this transition does take place, the state instantaneously resets to (q', x') where x' is determined by the reset map $r(q, q', x, \upsilon, \delta)$. Changes in υ and δ are discrete events that either *enable* a transition from q to q' by making sure $x \in \text{Guard}(q, q', \upsilon, \delta)$ or *force* a transition out of q by making sure $x \notin \text{Inv}(q, \upsilon, \delta)$. We will also use $\mathcal{E}$ to denote the set of all events that cause discrete state transitions and will classify events in a manner that suits the purposes of perturbation analysis.

In what follows, we describe the general framework for IPA presented in Ref. [10] and generalized in Ref. [13]. Let $\theta \in \Theta \subset \mathbb{R}^l$ be a global variable, henceforth called the *control parameter*, where Θ is a given compact, convex set. This may represent a system design parameter, a parameter of an input process, or a parameter that characterizes a policy used in controlling this system. The disturbance input $d \in D$ encompasses various random processes that affect the evolution of the state (q, x). We will assume that all such processes are defined over a common probability space, $(\Omega, \mathcal{F}, P)$. Let us fix a particular value of the parameter $\theta \in \Theta$ and study a resulting sample path of the SHS. Over such a sample path, let $\tau_k(\theta)$, $k = 1, 2, \ldots$, denote the occurrence times of the discrete events in increasing order, and define $\tau_0(\theta) = 0$ for convenience. We will use the notation τ_k instead of $\tau_k(\theta)$ when no confusion arises. The continuous state is also generally a function of θ, as well as of t, and is thus denoted by $x(\theta, t)$. Over an interval $[\tau_k(\theta), \tau_{k+1}(\theta))$, the system is at some mode during which the time-driven state satisfies

$$\dot{x} = f_k(x, \theta, t), \tag{1}$$

where $\dot{x}$ denotes $\partial x/\partial t$. Note that we suppress the dependence of f_k on the inputs $u \in U$ and $d \in D$ and stress instead its dependence on the parameter θ which may generally affect either u or d or both. The purpose of perturbation analysis is to study how changes in θ influence the state $x(\theta, t)$ and the event times $\tau_k(\theta)$ and, ultimately, how they influence interesting performance metrics which are generally expressed in terms of these variables. The following assumption guarantees that Eq. (1) has a unique solution w.p.1 for a given initial boundary condition $x(\theta, \tau_k)$ at time $\tau_k(\theta)$:

Assumption 1 W.p.1, there exists a finite set of points $t_j \in [\tau_k(\theta), \tau_{k+1}(\theta))$, $j = 1, 2, \ldots$, which are independent of θ, such that, the function f_k is continuously differentiable on $\mathbb{R}^n \times \Theta \times ([\tau_k(\theta), \tau_{k+1}(\theta)) \setminus \{t_1, t_2, \ldots\})$. Moreover, there exists a random number $K > 0$ such that $E[K] < \infty$ and the norm of the first derivative of f_k on $\mathbb{R}^n \times \Theta \times ([\tau_k(\theta), \tau_{k+1}(\theta)) \setminus \{t_1, t_2, \ldots\})$ is bounded from above by K.

An event occurring at time $\tau_{k+1}(\theta)$ triggers a change in the mode of the system, which may also result in new dynamics represented by f_{k+1}, although this may not always be the case; for example, two modes may be distinct because the state $x(\theta, t)$ enters a new region where the system's performance is measured differently without altering its time-driven dynamics (i.e., $f_{k+1} = f_k$). The event times $\{\tau_k(\theta)\}$ play an important role in defining the interactions between the time-driven and event-driven dynamics of the system.

We now classify events that define the set $\mathcal{E}$ as follows:

1. Exogenous events. An event is *exogenous* if it causes a discrete state transition at time τ_k independent of the controllable vector θ and satisfies $\frac{\mathrm{d}\tau_k}{\mathrm{d}\theta} = 0$. Exogenous events typically correspond to uncontrolled random changes in input processes.

2. Endogenous events. An event occurring at time τ_k is *endogenous* if there exists a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that

$$\tau_k = \min\{t > \tau_{k-1} \ : \ g_k(x(\theta, t), \theta) = 0\}. \tag{2}$$

The function g_k normally corresponds to a guard condition in a hybrid automaton model.

3. Induced events. An event at time τ_k is *induced* if it is triggered by the occurrence of another event at time $\tau_m \leqslant \tau_k$. The triggering event may be exogenous, endogenous, or itself an induced event. The events that trigger induced events are identified by a subset of the event set, $\mathcal{E}_I \subseteq \mathcal{E}$.

Consider a performance function of the control parameter θ:

$$J(\theta; x(\theta, 0), T) = E[\mathcal{L}(\theta; x(\theta, 0), T)],$$

where $\mathcal{L}(\theta; x(\theta, 0), T)$ is a sample function of interest evaluated in the interval $[0, T]$ with initial conditions $x(\theta, 0)$. For simplicity, we write $J(\theta)$ and $\mathcal{L}(\theta)$. Suppose that there are N events (independent of θ) occurring during the time interval $[0, T]$ and define $\tau_0 = 0$ and $\tau_{N+1} = T$. Let $L_k : \mathbb{R}^n \times \Theta \times \mathbb{R}^+ \to \mathbb{R}$ be a function satisfying Assumption 1 and define $\mathcal{L}(\theta)$ by

$$\mathcal{L}(\theta) = \sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t)\mathrm{d}t, \tag{3}$$

where we reiterate that $x = x(\theta, t)$ is a function of θ and t. We also point out that the restriction of the definition of $J(\theta)$ to a finite horizon T is made merely for the sake of simplicity of exposition.

Given that we do not wish to impose any limitations (other than mild technical conditions) on the random

processes that characterize the discrete or continuous disturbance inputs in our hybrid automaton model, it is infeasible to obtain closed-form expressions for $J(\theta)$. Therefore, for the purpose of optimization, we resort to iterative methods such as stochastic approximation algorithms (e.g., Ref. [15]) which are driven by estimates of the cost function gradient with respect to the parameter vector of interest. Thus, we are interested in estimating $\mathrm{d}J/\mathrm{d}\theta$ based on sample path data, where a sample path of the system may be directly observed or it may be obtained through simulation. We then seek to obtain θ^* minimizing $J(\theta)$ through an iterative scheme of the form

$$\theta_{n+1} = \theta_n - \eta_n H_n(\theta_n; x(\theta,0), T, \omega_n), \quad n = 0, 1, \ldots, \quad (4)$$

where $H_n(\theta_n; x(0), T, \omega_n)$ is an estimate of $\mathrm{d}J/\mathrm{d}\theta$ evaluated at θ_n and based on information obtained from a sample path denoted by ω_n and $\{\eta_n\}$ is an appropriately selected step size sequence. In order to execute an algorithm such as Eq. (4), we need the estimate $H_n(\theta_n)$ of $\mathrm{d}J/\mathrm{d}\theta$. The IPA approach is based on using the sample derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$ as an estimate of $\mathrm{d}J/\mathrm{d}\theta$. The strength of the approach is that $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$ can be obtained from observable sample path data alone and, usually, in a very simple manner that can be readily implemented on line. Moreover, it is often the case that $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$ is an *unbiased* estimate of $\mathrm{d}J/\mathrm{d}\theta$, a property that allows us to use Eq. (4) in obtaining θ^*. We will return to this issue later, and concentrate first on deriving the IPA estimates $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$.

2.1 Infinitesimal perturbation analysis

Let us fix $\theta \in \Theta$, consider a particular sample path, and assume for the time being that all derivatives mentioned in the sequel do exist. To simplify notation, we define the following for all state and event time sample derivatives:

$$x'(t) \equiv \frac{\partial x(\theta,t)}{\partial \theta}, \ \tau_k' \equiv \frac{\partial \tau_k}{\partial \theta}, \ k = 0, 1, \ldots, N. \quad (5)$$

In addition, we will write $f_k(t)$ instead of $f_k(x, \theta, t)$ whenever no ambiguity arises. By taking derivatives with respect to θ in Eq. (1) on the interval $[\tau_k(\theta), \tau_{k+1}(\theta))$ we get

$$\frac{\mathrm{d}}{\mathrm{d}t}x'(t) = \frac{\partial f_k(t)}{\partial x}x'(t) + \frac{\partial f_k(t)}{\partial \theta}. \quad (6)$$

The boundary (initial) condition of this linear equation is specified at time $t = \tau_k$, and by writing Eq. (1) in an integral form and taking derivatives with respect to θ when $x(\theta, t)$ is continuous in t at $t = \tau_k$, we obtain for $k = 1, 2, \ldots, N$:

$$x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)]\tau_k'. \quad (7)$$

We note that whereas $x(\theta, t)$ is often continuous in $t, x'(t)$ may be discontinuous in t at the event times τ_k, hence the left and right limits above are generally different. If $x(\theta, t)$ is not continuous in t at $t = \tau_k$, the value of $x(\tau_k^+)$ is determined by the reset function $r(q, q', x, \upsilon, \delta)$ discussed earlier and

$$x'(\tau_k^+) = \frac{\mathrm{d}r(q, q', x, \upsilon, \delta)}{\mathrm{d}\theta}. \quad (8)$$

Furthermore, once the initial condition $x'(\tau_k^+)$ is given, the linearized state trajectory $\{x'(t)\}$ can be computed in the interval $t \in [\tau_k(\theta), \tau_{k+1}(\theta))$ by solving Eq. (6) to obtain:

$$x'(t) = e^{\int_{\tau_k}^t \frac{\partial f_k(u)}{\partial x}\mathrm{d}u}\left[\int_{\tau_k}^t \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^t \frac{\partial f_k(u)}{\partial x}\mathrm{d}u}\mathrm{d}v + \xi_k\right] \quad (9)$$

with the constant ξ_k determined from $x'(\tau_k^+)$ in Eq. (7), since $x'(\tau_k^-)$ is the final-time boundary condition in the interval $[\tau_{k-1}(\theta), \tau_k(\theta))$, or it is obtained from Eq. (8).

Clearly, to complete the description of the trajectory of the linearized system (6) and (7), we have to specify the derivative τ_k' which appears in Eq. (7). Since τ_k, $k = 1, 2, \ldots$, are the mode-switching times, these derivatives explicitly depend on the interaction between the time-driven dynamics and the event-driven dynamics, and specifically on the type of event occurring at time τ_k. Using the event classification given earlier, we have the following.

1. Exogenous events. By definition, such events are independent of θ, therefore $\tau_k' = 0$.

2. Endogenous events. In this case, Eq. (2) holds and taking derivatives with respect to θ we get

$$\frac{\partial g_k}{\partial x}\left[x'(\tau_k^-) + f_k(\tau_k^-)\tau_k'\right] + \frac{\partial g_k}{\partial \theta} = 0, \quad (10)$$

which, assuming $\frac{\partial g_k}{\partial x}f_k(\tau_k^-) \neq 0$, can be rewritten as

$$\tau_k' = -\left[\frac{\partial g_k}{\partial x}f_k(\tau_k^-)\right]^{-1}\left(\frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial x}x'(\tau_k^-)\right). \quad (11)$$

3. Induced events. If an induced event occurs at $t = \tau_k$, the value of τ_k' depends on the derivative τ_m' where $\tau_m \leqslant \tau_k$ is the time when the associated triggering event takes place. The event induced at τ_m will occur at some time $\tau_m + \omega(\tau_m)$, where $\omega(\tau_m)$ is a random variable which is generally dependent on the continuous and discrete states $x(\tau_m)$ and $q(\tau_m)$ respectively. This implies the need for additional state variables, denoted by $y_m(\theta, t)$, $m = 1, 2, \ldots$, associated with events occurring at times τ_m, $m = 1, 2, \ldots$. The role of each such state variable is to provide a "timer" activated when a triggering event occurs. Recalling that triggering events are identified as belonging to a set $\mathcal{E}_I \subseteq \mathcal{E}$, let e_k denote the event occurring at τ_k and define $\mathcal{F}_k = \{m : e_m \in \mathcal{E}_I, m \leqslant k\}$ to be the set of all indices with corresponding triggering events up to τ_k. Omitting the dependence on

θ for simplicity, the dynamics of $y_m(t)$ are then given by

$$\dot{y}_m(t) = \begin{cases} -C(t), & \tau_m \leqslant t < \tau_m + \omega(\tau_m),\ m \in \digamma_m, \\ 0, & \text{otherwise}, \end{cases}$$
$$y_m(\tau_m^+) = \begin{cases} y_0, & y_m(\tau_m^-) = 0,\ m \in \digamma_m, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where y_0 is an initial value for the timer $y_m(t)$ which decreases at a "clock rate" $C(t) > 0$ until $y_m(\tau_m + \omega(\tau_m)) = 0$ and the associated induced event takes place. Clearly, these state variables are only used for induced events, so that $y_m(t) = 0$ unless $m \in \digamma_m$. The value of y_0 may depend on θ or on the continuous and discrete states $x(\tau_m)$ and $q(\tau_m)$, while the clock rate $C(t)$ may depend on $x(t)$ and $q(t)$ in general, and possibly θ. However, in most simple cases where we are interested in modeling an induced event to occur at time $\tau_m + \omega(\tau_m)$, we have $y_0 = \omega(\tau_m)$ and $C(t) = 1$, i.e., the timer simply counts down for a total of $\omega(\tau_m)$ time units until the induced event takes place. An example where y_0 in fact depends on the state $x(\tau_m)$ and the clock rate $C(t)$ is not necessarily constant arises in the case of multiclass resource contention systems as described in Ref. [16]. Henceforth, we will consider $y_m(t),\ m = 1, 2, \ldots$, as part of the continuous state of the SHS and, similar to Eq. (5), we set

$$y'_m(t) \equiv \frac{\partial y_m(t)}{\partial \theta}, \quad m = 1, 2, \ldots, N. \tag{13}$$

For the common case where y_0 is independent of θ and $C(t)$ is a constant $c > 0$ in Eq. (12), the following lemma facilitates the computation of τ'_k for an induced event occurring at τ_k. Its proof is given in Ref. [13].

Lemma 2.1 If in Eq. (12) y_0 is independent of θ and $C(t) = c > 0$ (constant), then $\tau'_k = \tau'_m$.

With the inclusion of the state variables $y_m(t),\ m = 1, 2, \ldots, N$, the derivatives $x'(t)$, τ'_k, and $y'_m(t)$ can be evaluated through Eqs. (6) – (11). In general, this evaluation is recursive over the event (mode switching) index $k = 0, 1, \ldots$. In some cases, however, it can be reduced to simple expressions, as seen in the analysis of many SFMs, e.g., Ref. [5].

Remark If an SHS does not involve induced events and if the state does not experience discontinuities when a mode-switching event occurs, then the full extent of IPA reduces to three equations:

1) Equation (9), which describes how the state derivative $x'(t)$ evolves over $[\tau_k(\theta), \tau_{k+1}(\theta))$;

2) Equation (7), which specifies the initial condition ξ_k in Eq. (9); and

3) Either $\tau'_k = 0$ or Eq. (11) depending on the event type at $\tau_k(\theta)$, which specifies the event time derivative present in Eq. (7).

Now the IPA derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$ can be obtained by taking derivatives in Eq. (3) with respect to θ:

$$\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta} = \sum_{k=0}^{N} \frac{\mathrm{d}}{\mathrm{d}\theta} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t)\mathrm{d}t. \tag{14}$$

Applying the Leibnitz rule we obtain, for every $k = 0, 1, \ldots, N$,

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\theta} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t)\mathrm{d}t \\ &= \int_{\tau_k}^{\tau_{k+1}} \left[\frac{\partial L_k}{\partial x}(x, \theta, t)x'(t) + \frac{\partial L_k}{\partial \theta}(x, \theta, t) \right] \mathrm{d}t \\ &\quad + L_k(x(\tau_{k+1}), \theta, \tau_{k+1})\tau'_{k+1} - L_k(x(\tau_k), \theta, \tau_k)\tau'_k, \end{aligned} \tag{15}$$

where $x'(t)$ and τ'_k are determined through Eqs. (6) – (11). What makes IPA appealing, especially in the SFM setting, is the simple form the right-hand-side above often assumes.

We close this section with a comment on the unbiasedness of the IPA derivative $\mathrm{d}\mathcal{L}/\mathrm{d}\theta$. This IPA derivative is statistically unbiased [2,3] if, for every $\theta \in \Theta$,

$$E\left[\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta}\right] = \frac{\mathrm{d}}{\mathrm{d}\theta} E[\mathcal{L}(\theta)] = \frac{\mathrm{d}J(\theta)}{\mathrm{d}\theta}. \tag{16}$$

The main motivation for studying IPA in the SHS setting is that it yields unbiased derivatives for a large class of systems and performance metrics compared to the traditional DES setting [2]. The following conditions have been established in Ref. [17] as sufficient for the unbiasedness of IPA:

Proposition 2.1 Suppose that the following conditions are in force: 1) For every $\theta \in \Theta$, the derivative $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ exists w.p.1. 2) W.p.1, the function $\mathcal{L}(\theta)$ is Lipschitz continuous on Θ, and the Lipschitz constant has a finite first moment. Fix $\theta \in \Theta$. Then, the derivative $\mathrm{d}J(\theta)/\mathrm{d}\theta$ exists, and the IPA derivative $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ is unbiased.

The crucial assumption for Proposition 2.1 is the continuity of the sample performance function $\mathcal{L}(\theta)$, which in many SHS (and SFMs in particular), such continuity is guaranteed in a straightforward manner. Differentiability w.p.1 at a given $\theta \in \Theta$ often follows from mild technical assumptions on the probability law underlying the system, such as the exclusion of co-occurrence of multiple events (see Ref. [16]). Lipschitz continuity of $\mathcal{L}(\theta)$ generally follows from upper boundedness of $|\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta|$ by an absolutely integrable random variable, generally a weak assumption. In light of these observations, the proofs of unbiasedness of IPA have become standardized and the assumptions in Proposition 2.1 can be verified fairly easily from the context of a particular problem.

3 Some IPA properties

In this section, we derive some properties of IPA within the general framework of the previous section leading to sufficient conditions under which IPA becomes particularly simple and efficient to implement with minimal information required about the underlying SHS dynamics.

The first question we address is related to $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ in Eq. (14), which, as seen in Eq. (15), generally depends on information accumulated over all $t \in [\tau_k, \tau_{k+1})$. It is, however, often the case that it depends *only* on information related to the event times τ_k, τ_{k+1}, resulting in an IPA estimator which is very simple to implement. Using the notation $L'_k(x,t,\theta) \equiv \frac{\partial L_k(x,t,\theta)}{\partial \theta}$, we can rewrite $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ in Eq. (14) as

$$\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta} = \sum_k \Big[\tau'_{k+1} L_k\left(\tau^+_{k+1}\right) - \tau'_k L_k\left(\tau^+_k\right) + \int_{\tau_k}^{\tau_{k+1}} L'_k(x,t,\theta)\,\mathrm{d}t\Big]. \tag{17}$$

The following lemma provides two sufficient conditions under which $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ is independent of t and involves only the event time derivatives τ'_k, τ'_{k+1} and the "local" performance $L_k\left(\tau^+_{k+1}\right), L_k\left(\tau^+_k\right)$ which is obviously easy to observe.

Lemma 3.1 If condition 1) or 2) below holds, then $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ depends only on information available at event times $\{\tau_k\}$, $k = 0, 1, \ldots$.

1) $L_k(x,t,\theta)$ is independent of t over $[\tau_k, \tau_{k+1})$ for all $k = 0, 1, \ldots$;

2) $L_k(x,t,\theta)$ is only a function of x and the following condition holds for all $t \in [\tau_k, \tau_{k+1}), k = 0, 1, \ldots$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L_k}{\partial x} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial f_k}{\partial x} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial f_k}{\partial \theta} = 0. \tag{18}$$

Proof Under condition 1), $L'_k(x,t,\theta)$ is also obviously independent of t and the integral term in Eq. (17) becomes $L'_k(x,\theta)(\tau_{k+1} - \tau_k)$, which only uses information at event times τ_k and τ_{k+1}. Therefore, $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ requires information at event times only.

Under condition 2), $\partial L_k/\partial x$, $\partial f_k/\partial x$ and $\partial f_k/\partial \theta$ are constant between consecutive events, which we will denote as

$$\frac{\partial L_k}{\partial x} = C_{k,1}, \quad \frac{\partial f_k}{\partial x} = C_{k,2}, \quad \frac{\partial f_k}{\partial \theta} = C_{k,3}.$$

Recalling the dynamics of $x'(t)$ given in Eq. (6), and in particular using Eq. (9), we have

$$x'(t) = e^{C_{k,2}(t-\tau_k)}\left[\frac{C_{k,3}e^{C_{k,2}\tau_k}}{C_{k,2}} \cdot \left(e^{-C_{k,2}\tau_k} - e^{-C_{k,2}t}\right) + x'\left(\tau^+_k\right)\right],$$

for all $t \in [\tau_k, \tau_{k+1})$. Using the above two equations,

$$\begin{aligned}\int_{\tau_k}^{\tau_{k+1}} & L'_k(x,t,\theta)\,\mathrm{d}t \\ &= \int_{\tau_k}^{\tau_{k+1}} C_{k,1}x'(t)\,\mathrm{d}t \\ &= C_{k,1}e^{-C_{k,2}\tau_k}\left[\left(\frac{C_{k,3}}{C_{k,2}} + \frac{x'\left(\tau^+_k\right)}{C_{k,2}}\right)\right. \\ &\quad \left.\cdot\left(e^{C_{k,2}\tau_{k+1}} - e^{C_{k,2}\tau_k}\right) - \frac{C_{k,3}e^{C_{k,2}\tau_k}}{C_{k,2}}(\tau_{k+1} - \tau_k)\right],\end{aligned} \tag{19}$$

from which we can see that the integral term in Eq. (17) depends only on information at event times τ_k and τ_{k+1}. Therefore, the overall evaluation of $\mathrm{d}\mathcal{L}(\theta)/\mathrm{d}\theta$ will also only require information at event times. ■

The second question we address is related to the discontinuity in $x'(t)$ at event times, described in Eq. (7). This happens when endogenous events occur, since for exogenous events we have $\tau'_k = 0$. The next lemma identifies a simple condition under which $x'\left(\tau^+_k\right)$ is independent of the dynamics f before the event at τ_k. This implies that we can evaluate the sensitivity of the state with respect to θ without any knowledge of the state trajectory in the interval $[\tau_{k-1}, \tau_k)$ prior to this event. Moreover, under an additional condition, we obtain $x'\left(\tau^+_k\right) = 0$, implying that the effect of θ is "forgotten" and one can reset the perturbation process. This allows us to study the SHS over reset cycles, greatly simplifying the IPA process.

Lemma 3.2 Suppose an endogenous event occurs at τ_k with switching function $g(x,\theta)$. If $f_k(\tau^+_k) = 0$, $x'(\tau^+_k)$ is independent of f_{k-1}. If, in addition, $\frac{\partial g}{\partial \theta} = 0$, then $x'(\tau^+_k) = 0$.

Proof Using Eqs. (7) and (11), we have

$$x'\left(\tau^+_k\right) = \left(\frac{\partial g}{\partial x}\right)^{-1}\left[\frac{\partial g}{\partial \theta} + \frac{f_k\left(\tau^+_k\right)}{f_{k-1}\left(\tau^-_k\right)}\frac{\partial g}{\partial x}x'\left(\tau^-_k\right) - \frac{f_k\left(\tau^+_k\right)}{f_{k-1}\left(\tau^-_k\right)}\frac{\partial g}{\partial \theta}\right]. \tag{20}$$

It is easy to see that when $f_k\left(\tau^+_k\right) = 0, x'\left(\tau^+_k\right) = \frac{\partial g}{\partial \theta}\left(\frac{\partial g}{\partial x}\right)^{-1}$ which is independent of f_{k-1}. Furthermore, if $\frac{\partial g}{\partial \theta} = 0$, then $x'\left(\tau^+_k\right) = 0$. ■

The condition $f_k\left(\tau^+_k\right) = 0$ typically indicates a saturation effect or the state reaching a boundary that cannot be crossed, e.g., when the state is constrained to be non-negative. When the conditions in the two lemmas are satisfied, IPA provides sensitivity estimates that do not require knowledge of the noise processes or the detailed time-driven dynamics of the system, other than mild technical conditions. Thus, one need not have a

detailed model (captured by f_{k-1}) to describe the state behavior through $\dot{x} = f_{k-1}(x, \theta, t)$, $t \in [\tau_{k-1}, \tau_k)$ in order to estimate the effect of θ on this behavior. This explains why simple abstractions of a complex stochastic system are often adequate to perform sensitivity analysis and optimization, as long as the event times corresponding to discrete state transitions are accurately observed and the local system behavior at these event times, e.g., $x'\left(\tau_k^+\right)$ in Eq. (19), can also be measured or calculated. In the case of SFMs, the conditions in these lemmas are frequently satisfied since 1) common performance metrics such as workloads or overflow rates satisfy Eq. (18), and 2) flow systems involve non-negative continuous states and are constrained by capacities that give rise to dynamics of the form $\dot{x} = 0$.

4 General scheme for abstracting DES to SFM

As mentioned in the introduction, one of the main motivations for SHS is to use them as abstractions of complex DES, where the dynamics are abstracted to an appropriate level that enables effective optimization and control of the underlying DES. In this section, we propose a general scheme to abstract a DES to an HS, under the assumption that the state of the DES is represented by integers; this is typically the case for queueing systems, hence the abstracted SHS is usually an SFM. As an example, we apply the proposed scheme to $G/G/1/K$ systems, and obtain the corresponding SFMs which can be seen to be the same as those obtained in prior work without this systematic framework.

Consider a DES modeled by an automaton $G = \{X, E, f, \Gamma\}$, where $X \subseteq \mathbb{R}^n$ is the set of *states*, E is the (finite) set of *events* associated with G, $f : X \times E \to X$ is the *transition function*, i.e., $f(x, e) = y$ means that there is a transition caused by event e from state x to state y, and $\Gamma : X \to 2^E$ is the *active event function*, i.e., $\Gamma(x)$ is the set of all events e for which $f(x, e)$ is defined and referred to as the "active event set" of G at x (see also Ref. [2]). In addition, for any $x \in X$, $e \in E$, we define a function $h_{x,e} : X \to X$, such that $h_{x,e}(x) = f(x, e)$.

In what follows, we present a general scheme to abstract G to a hybrid automaton modeling an HS. There are two steps in the scheme: the first step is to partition states in X into a number of discrete aggregate states; then, in the second step, the discrete transitions within each aggregate state are abstracted into continuous (time-driven) dynamics.

Step 1 A partition divides the set of states X into non-overlapping and non-empty subsets and is described by the partition function $\mathcal{P} : X \to \mathbb{Z}$, such that $\mathcal{P}(a) = \mathcal{P}(b)$ if and only if states a and b are in the same subset. A partition $\mathcal{P}_1$ is said to be *larger* than $\mathcal{P}_2$ if any states in the same subset in $\mathcal{P}_2$ are also in the same subset in $\mathcal{P}_1$, i.e.,

$$\mathcal{P}_2(a) = \mathcal{P}_2(b) \Rightarrow \mathcal{P}_1(a) = \mathcal{P}_1(b).$$

We also define *interior states* to be states $x \in X$ that have no active events causing transitions to states outside the subset they are in, i.e., a is said to be an interior state of $\mathcal{P}$ if for all events $e \in \Gamma(a)$, $\mathcal{P}(f(a, e)) = \mathcal{P}(a)$. States that are not interior are termed *boundary states*.

The partitions that we are interested in for the purpose of abstracting G should satisfy the following two criteria:

Criterion 1 For two states $a, b \in X$, if $\mathcal{P}(a) = \mathcal{P}(b)$, then the following holds:

$$\Gamma(a) = \Gamma(b). \tag{21}$$

Criterion 2 For any two states $a, b \in X$, if $\mathcal{P}(a) = \mathcal{P}(b)$, then for any event $e \in \Gamma(a) \cap \Gamma(b)$ such that $\mathcal{P}(f(a, e)) = \mathcal{P}(f(b, e)) = \mathcal{P}(a)$ the following holds:

$$h_{a,e} = h_{b,e}, \tag{22}$$

where $h_{a,e}, h_{b,e}$ are two functions in the state space as defined above, i.e., they satisfy $h_{a,e}(a) = f(a, e)$ and $h_{b,e}(b) = f(b, e)$. Condition (22) states an equivalence relation between two functions such that for all $x \in X$, $h_{a,e}(x) = h_{b,e}(x)$ holds. For example, $h_{a,e}(x) = h_{b,e}(x) = x + 1$; or $h_{a,e}(x)$, $h_{b,e}(x)$ are constant functions such as $h_{a,e}(x) = h_{b,e}(x) = 0$.

The first criterion indicates that all states $x \in X$ in the same aggregate state (subset) must have the same active event set. The second criterion simply states that an active event of an aggregate state will have the same effect for all the interior states of that aggregate state. The partition we are seeking is the largest one satisfying the above two criteria.

Step 2 After the state partition is carried out, the next step is to abstract discrete transitions within each of the aggregate states to some form of continuous dynamics. This is achieved by analyzing the effects of all active events in changing the values of the interior states. For example, for an interior state $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ in a certain aggregate state, an active event e_1 at x that increases the value of x_i can be abstracted as a continuous inflow with a rate $\alpha_i(t)$, while an active event e_2 that decreases x_i can be abstracted as a continuous outflow with a rate $\beta_i(t)$, where x_i is viewed as a "flow content" in the system. Doing so for all events in the active set gives flow-like continuous dynamics for all elements of state x, such that

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \alpha_i(t) - \beta_i(t).$$

Finally, we get the HS abstraction where the event-driven part is represented by the aggregate states

obtained in Step 1 and events that cause transitions among them, and the time-driven part is represented by the continuous system evolution within each of the aggregate states obtained through Step 2. In addition, the stochastic characteristics of the original DES remain in the abstraction model, so that events between aggregate states may occur randomly, and the flow rates in the continuous dynamics within each aggregate state may be stochastic processes. Note that the abstraction process aims at obtaining the structure of a hybrid automaton and not the explicit values of the flow processes involved.

As previously mentioned, we limit ourselves to DES with integer-valued state variables, hence the abstracted continuous dynamics normally describe flows, leading to the class of SFMs as the abstracted SHS. For more general DES, the abstraction process is more complicated and may generate more general SHS other than SFM.

4.1 Example: $G/G/1/K$ system

In order to illustrate the general scheme described above, we apply it to a single-class single-server queueing system with finite capacity, i.e., a $G/G/1/K$ queueing system with a first come first serve (FCFS) serving policy. Let $x \in \{0, 1, \ldots, K\}$ be the state of the system, representing the number of jobs in the queue. The queue has a capacity K, so that when $x = K$ further incoming jobs will be blocked. There are two types of events in this system, i.e., $E = \{a, d\}$, where a stands for acceptance of a job arrival, and d represents a departure of a job after being processed. Note that when event a occurs at $x = K$, this arrival is blocked because the queue is full. The automaton model of this system is shown in Fig. 1.

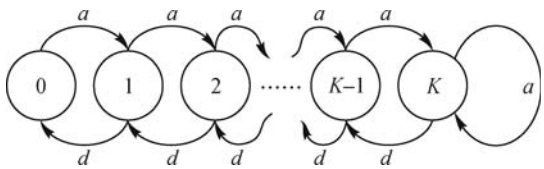

Fig. 1 Automaton model of $G/G/1/K$ system

Applying the general scheme of the previous section, we divide the state space of the $G/G/1/K$ system into the following three aggregate states:

Aggregate State 1 This aggregate state includes only $x = 0$, and the active event set is obviously $\Gamma(0) = \{a\}$.

Aggregate State 2 x belongs to this aggregate state if $0 < x < K$, and the active event set is $\{a, d\}$, since there can be both job arrivals and departures, and all arrivals will be accepted given that the queue has not reached its capacity. It is also easy to check that criterion (22) is satisfied, as event a increases all x by 1, while event d decreases all x by 1.

Aggregate State 3 This aggregate state includes only $x = K$, and the active event set is $\Gamma(K) = \{a, d\}$. However, unlike the previous aggregate state, arrivals at this state will be blocked because the queue is full.

Next, we abstract the transitions within the above three aggregate states to appropriate continuous dynamics. First, for aggregate states 1 and 3, since they are both singletons with no discrete transitions within, the corresponding continuous dynamics are obviously $\dot{x} = 0$. Within aggregate state 2, the active discrete transitions are job arrivals and job departures, which can be both abstracted as continuous flows, thus giving the continuous dynamics $\dot{x} = \alpha(t) - \beta(t)$, where $\alpha(t)$ is the inflow rate that corresponds to job arrivals, and $\beta(t)$ is the outflow rate that comes from job departures.

It follows from the above analysis that the SHS abstraction of the $G/G/1/K$ system is an SFM with dynamics given by

$$\dot{x} = \begin{cases} 0, & x = 0 \text{ and } \alpha(t) \leqslant \beta(t), \\ 0, & x = K \text{ and } \alpha(t) \geqslant \beta(t), \\ \alpha(t) - \beta(t), & \text{otherwise,} \end{cases}$$

consistent with what has been presented in previous related work [5]. Note that there are some conditions in the above dynamics, e.g., $\alpha(t) \leqslant \beta(t)$, that do not directly follow from the two steps in the abstraction scheme. They are included simply to ensure flow conservation.

5 Resource contention games

As mentioned in the Introduction, one of the recent advances in IPA for SHS is the ability to develop unbiased performance gradient estimates in multi-agent settings where each agent has its own objective function. This allows us to analyze a variety of game situations. In particular, we consider *resource contention games*, a class of non-cooperative games in which two or more "users" compete for one or more sharable resources by submitting requests for its use over time. The traditional server-queue modeling paradigm gives rise to various types of DES depending on the types of users and resources involved and features such as user priorities, scheduling disciplines, and pricing mechanisms. As already argued, however, these DES become increasingly complex with large volumes of user requests for a resource and with elaborate scheduling policies. SFMs provide an abstraction process which has proved useful in analyzing such problems. Most of the problems dealt with in this context adopt a purely "system-centric" point of view: the system defines an objective function and seeks to optimize it through appropriate control actions. However, when there are multiple user types (also referred to as user "classes"), each user may define its own objective function and seek to optimize it.

This gives rise to a game setting with a "user-centric" optimization perspective.

In this section, we will present a general setting for resource contention games modeled through SFMs and describe how IPA may be applied to estimate performance metric derivatives that are then used in gradient-based mechanisms for both "system-centric" and "user-centric" optimization. To accomplish this goal, we need to model SFMs with multiple user classes and accommodate mechanisms such as FCFS policy which, while very simple in the DES context, is not easy to capture in the flow paradigm. Until recently, IPA had been applied to SFMs where flows are differentiated in terms of admission to a system, but once admitted all flows are treated alike, e.g., Ref. [18]. IPA for SFMs that can differentiate flow classes in terms of how they are processed by a resource requires class-dependent performance metrics and gradient estimates of such metrics with respect to controllable parameters. Chen et al. [19] studied a multiclass SFM to analyze a dynamic priority call center and introduced a model differentiating among flow classes even after they enter the system; however, the analysis is very specific to the call center application. The general framework presented in the previous sections, and the use of induced events in particular, has enabled the use of IPA for class-dependent performance metrics (see Refs. [16,20]). This opens up a new frontier in the analysis of SFMs and IPA, allowing us to study the difference between *user-centric* and *system-centric* optimization, and place resource contention problems in a non-cooperative game framework.

In what follows, we will formulate a general resource contention game in the multiclass SFM context developed in Ref. [21] and show how the IPA framework of the previous section can be applied to this problem. To illustrate the approach, we will solve a simple specific instance of the general problem and provide some numerical results.

5.1 SFM for resource contention games

Suppose there are N "players" corresponding to N user classes competing for a resource with total service capacity $C(t)$, which can 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 $c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t))$, which satisfies for all $i = 1, 2, \ldots, N$:

$$\sum_{i=1}^{N} c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t)) = C(t),$$
$$0 \leqslant c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t)) \leqslant C(t), \tag{23}$$

where $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$, $x_i(t) \geqslant 0$ is the amount of the ith user class requests accumulated in the system at time t, and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$ is a vector of control parameters used to determine how the resource capacity is allocated to different classes, hence affecting the system dynamics as well, which are given by

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t^+} = f_i(\boldsymbol{x}(t), \boldsymbol{\theta}) = \begin{cases} 0, \\ \quad x_i(t) = 0 \text{ and } \alpha_i(t) - \beta_i(\boldsymbol{x}(t), \boldsymbol{\theta}) \leqslant 0, \\ \alpha_i(t) - \beta_i(\boldsymbol{x}(t), \boldsymbol{\theta}), \\ \quad \text{otherwise}, \end{cases} \tag{24}$$

where $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ is a control policy determining the actual service rate of class $i = 1, 2, \ldots, N$ and satisfying:

$$\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta}) \leqslant c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t)) \quad \text{for all } t, \tag{25}$$
$$\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta}) = 0 \quad \text{for all } t \text{ s.t. } x_i(t) = 0. \tag{26}$$

In addition, we assume that $\alpha_i(t)$ is independent of $\boldsymbol{\theta}$. Note that, due to Eq. (26), it follows from Eq. (24) that $\alpha_i(t) = \beta_i(\boldsymbol{x}(t), \boldsymbol{\theta}) = 0$ as long as $x_i(t) = 0$ and an "empty period" for class i ends whenever there is a change from $\alpha_i(t) = 0$ to $\alpha_i(t) > 0$. Clearly, the choice of $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ may affect whether "chattering" at $x_i = 0$ occurs or not.

Turning our attention to objective functions of interest in a resource contention game, we consider three metrics: 1) the *workload* that a user class accumulates and would like to minimize, 2) the resource *utilization* that each class would like to maximize, and 3) the *cost of resource usage* imposed by the system that each class would like to minimize. This general-purpose tradeoff is expressed through the following cost function for user class i:

$$J_i(\boldsymbol{\theta}) = E[\mathcal{L}_i(\boldsymbol{\theta})] = E\Big[\frac{1}{T}\int_0^T [x_i(t) + R_i \cdot \mathbf{1}[x_i(t) = 0] + p_i c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t))]\mathrm{d}t\Big], \tag{27}$$

where p_i is the unit resource price for user class i, and R_i is a given cost per unit time penalizing user class i while it remains idle ($\mathbf{1}[\cdot]$ is the usual indicator function). From the system's perspective, the minimization problem of interest involves a (possibly weighted) sum of all $J_i(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} w_i J_i(\boldsymbol{\theta}), \tag{28}$$

where w_i are non-negative weight parameters.

Using the notation established in the previous sections, let us classify the events that occur in this system as follows, where we shall use τ_k to denote the kth event time:

1. *Exogenous events.* In this particular system, the processes which are independent of $\boldsymbol{\theta}$ are the user

request rates (inflows) $\{\alpha_i(t)\}$ and the capacity $\{C(t)\}$. If these processes involve discontinuities, then any event associated with such a discontinuity is, by definition, an exogenous event.

2. *Endogenous events.* By the definition in Eq. (2), the event "$x_i(t)$ becomes 0" is an endogenous event with corresponding switching function

$$g(\boldsymbol{x}(t), \boldsymbol{\theta}) = x_i. \tag{29}$$

In addition, there may be other endogenous events depending on the specific nature of the control policies $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$. If there is a discontinuity in $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ for $x_i(t) > 0$ that causes a change in the time-driven dynamics $f_i(\boldsymbol{x}(t), \boldsymbol{\theta})$, then an endogenous event is defined with some associated switching function.

3. *Induced events.* Induced events are possible as shown in some previous work [16,20] where it is necessary to model a system treating user requests on an FCFS basis.

In what follows, we proceed without attempting to select any specific control policies $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ or $c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t))$ and derive event time and state derivatives (sensitivities) with respect to $\boldsymbol{\theta}$ based on the IPA framework presented in the previous sections.

First, for any exogenous event at τ_k, by definition, $\tau'_{k,j} = \frac{\partial \tau_k}{\partial \theta_j} = 0$, and $x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-)$. On the other hand, for an endogenous event "$x_i(t)$ becomes 0" using Eqs. (11) and (29), we have

$$\begin{aligned} \tau'_{k,j} &= -\frac{x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - \beta_i(\boldsymbol{x}(\tau_k^-), \boldsymbol{\theta})} \\ &= \frac{x'_{i,j}(\tau_k^-)}{\beta_i(\boldsymbol{x}(\tau_k^-), \boldsymbol{\theta})}, \end{aligned} \tag{30}$$

where we have used the fact that $\alpha_i(\tau_k^-) = 0$ based on Eqs. (24) and (26). We also observe that $f_i(\boldsymbol{x}(t), \boldsymbol{\theta}) = 0$ during a period with $x_i(t) = 0$ as seen in Eq. (24) and that $g(\boldsymbol{x}(t), \boldsymbol{\theta}) = x_i$, which satisfies both conditions of Lemma 3.2, therefore,

$$x'_{i,j}(\tau_k^+) = 0. \tag{31}$$

For any endogenous event at τ_k that causes a discontinuity in $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ when $x_i(t) > 0$, the event time derivatives $\tau'_{k,j}$ depend on the explicit expression for the associated switching function. Once this is available, we can derive $\tau'_{k,j}$ from Eq. (11). Then, using Eq. (7), $x'_{i,j}(\tau_k^+)$ is obtained as

$$\begin{aligned} x'_{i,j}(\tau_k^+) &= x'_{i,j}(\tau_k^-) + [f_i(\tau_k^-) - f_i(\tau_k^+)]\tau'_{k,j} \\ &= x'_{i,j}(\tau_k^-) + [\beta_i(\boldsymbol{x}(\tau_k^-), \boldsymbol{\theta}) \\ &\quad - \beta_i(\boldsymbol{x}(\tau_k^+), \boldsymbol{\theta})]\tau'_{k,j}. \end{aligned} \tag{32}$$

If induced events are present, recall that the SHS model must include the additional state variables $y_m(t)$ in Eq. (12). The precise way in which induced events occur is part of the specific process we are interested in which we do not address here in order to maintain the desired level of generality. However, a particular resource contention model involving induced events is studied in Ref. [16].

We now return to the ith sample function $\mathcal{L}_i(\boldsymbol{\theta})$ in Eq. (27) and rewrite it as

$$\begin{aligned} \mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{T}\sum_{k=0}^{N_T}\int_{\tau_k}^{\tau_{k+1}} [x_i(t) + p_i c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t))]\mathrm{d}t \\ + \frac{R_i}{T}\sum_{k\in\Phi_i}(\tau_{k+1} - \tau_k), \end{aligned}$$

where N_T is the number of events contained in $[0, T]$ and Φ_i is the set of all "empty periods" of class i, defined as

$$\Phi_i = \{k, \text{ s.t. } x_i(t) = 0 \text{ for all } t \in [\tau_k, \tau_{k+1})\}. \tag{33}$$

The IPA derivative $\partial\mathcal{L}_i/\partial\theta_j$ can be obtained as

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \theta_j} = \frac{1}{T}\sum_{k=0}^{N_T}\int_{\tau_k}^{\tau_{k+1}} \Bigg[x'_{i,j}(t) + p_i \frac{\partial c_i}{\partial \theta_j} \\ + p_i \sum_{m=1}^{N} \frac{\partial c_i}{\partial x_m} x'_{m,j}(t) \Bigg] \mathrm{d}t \\ + \frac{R_i}{T}\sum_{k\in\Phi_i}(\tau'_{k+1,j} - \tau'_{k,j}), \end{aligned} \tag{34}$$

where $\tau'_{k,j}$ are obtained as described above and $x'_{i,j}(t)$ are obtained by using Eq. (9), which in this case becomes

$$\begin{aligned} x'_{i,j}(t) = e^{-\int_{\tau_k}^{t} \frac{\partial \beta_i(\boldsymbol{x}(u), \boldsymbol{\theta})}{\partial x_i} \mathrm{d}u} \Bigg[\int_{\tau_k}^{t} \left(-\frac{\partial \beta_i(\boldsymbol{x}(v), \boldsymbol{\theta})}{\partial \theta_j} \right) \\ \cdot e^{\int_{\tau_k}^{t} \frac{\partial \beta_i(\boldsymbol{x}(u), \boldsymbol{\theta})}{\partial x_i} \mathrm{d}u} \mathrm{d}v + l \Bigg] \end{aligned} \tag{35}$$

for $t \in [\tau_k, \tau_{k+1})$, where l is an initial condition obtained through Eq. (32) or similarly, depending on any additional endogenous or induced events included in the system model. In addition, recalling the system dynamics in Eq. (26), $\beta_i = 0$ during periods $[\tau_k, \tau_{k+1})$ when $x_i(t) = 0$, therefore, using Eq. (35), $x'_{i,j}(t) = x'_{i,j}(\tau_k^+) = 0$ according to Eq. (31) for all $t \in [\tau_k, \tau_{k+1})$ and Eq. (34) reduces to

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \theta_j} = \frac{1}{T}\Bigg[\sum_{k\notin\Phi_i}\int_{\tau_k}^{\tau_{k+1}} x'_{i,j}(t)\,\mathrm{d}t + p_i \frac{\partial c_i}{\partial \theta_j}(\tau_{k+1} - \tau_k) \\ + \sum_{k=0}^{N_T}\left(p_i \int_{\tau_k}^{\tau_{k+1}} \sum_{m=1}^{N} \frac{\partial c_i}{\partial x_m} x'_{m,j}(t)\,\mathrm{d}t \right) \\ + R_i \sum_{k\in\Phi_i}(\tau'_{k+1,j} - \tau'_{k,j}) \Bigg]. \end{aligned} \tag{36}$$

Observe that $\partial c_i/\partial x_m, \partial c_i/\partial \theta_j$ are readily evaluated based on a given policy $c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t))$. Thus, the evaluation of $\mathrm{d}\mathcal{L}_i/\mathrm{d}\theta_j$ above depends on information

related to the event times τ_k and on $x'_{i,j}(t)$. If, however, the conditions of Lemma 3.1 are satisfied, then the entire expression depends only on information related to the event times τ_k. As an example, suppose $\beta_i(\boldsymbol{x}(t),\boldsymbol{\theta}) = C(t)\theta_i$ is a simple fixed resource allocation policy over some interval $[\tau_k,\tau_{k+1})$ and we have $f_{i,k}(t,\boldsymbol{\theta}) = \alpha_i(t) - C(t)\theta_i$. Then, it is easy to check that Eq. (18) in Lemma 3.1 holds, and we have $\frac{\partial\beta_i(\boldsymbol{x}(v),\boldsymbol{\theta})}{\partial\theta_j} = \mathbf{1}[i=j]$, $\frac{\partial\beta_i(\boldsymbol{x}(u),\boldsymbol{\theta})}{\partial x_i} = 0$ which reduces Eq. (35) for all $t\in[\tau_k,\tau_{k+1})$ to

$$x'_{i,j}(t) = x'_{i,j}(\tau_k^+) - \mathbf{1}[i=j]C(\tau_k)(t-\tau_k). \tag{37}$$

This is easy to compute and only requires directly observable event time information.

5.2 A specific resource contention game

In this section, we study a relatively simple specific instance of the general model, in order to illustrate how this general IPA framework can be directly applied and how to contrast a system-centric to a user-centric optimization solution. We note that resource contention games analyzed in Refs. [16,20] include induced events and specific control policies and were developed prior to the general IPA framework in Sect. 2.

Consider a game with N user classes and total service capacity $C(t)$ which is allocated to each class as follows:

$$c_i(\boldsymbol{x}(t),\boldsymbol{\theta},C(t)) = C(t)\theta_i,\quad i=1,2,\ldots,N, \tag{38}$$

with $\sum_{i=1}^N\theta_i = 1, 0\leqslant\theta_i\leqslant 1$. The system dynamics are given by Eq. (24), where the actual service rate policy $\beta_i(\boldsymbol{x}(t),\boldsymbol{\theta})$ is specified for $i=1,2,\ldots,N$ as

$$\beta_i(\boldsymbol{x}(t),\boldsymbol{\theta}) = \begin{cases} \alpha_i(t), & 0<\alpha_i(t)\leqslant C(t)\theta_i \\ & \quad\text{and } x_i(t)=\kappa_i, \\ \beta_{i,\min}, & 0<x_i(t)<\kappa_i, \\ 0, & x_i(t)=0, \\ C(t)\theta_i, & \text{otherwise.} \end{cases} \tag{39}$$

Under this policy, $\beta_i(\boldsymbol{x}(t),\boldsymbol{\theta}) = C(t)\theta_i$ as long as $x_i(t)\geqslant\kappa_i$, a given parameter for each class. If $0<x_i(t)<\kappa_i$ the policy guarantees class i a minimum service rate $\beta_{i,\min} < C(t)\theta_i$. To prevent "chattering" at $x_i=0$, we will assume that there exists $\bar{\alpha}_i>0$ such that $\bar{\alpha}_i = \inf_{t\geqslant 0}\{\alpha_i(t)\}$ and

$$\bar{\alpha}_i > \beta_{i,\min}. \tag{40}$$

If $\alpha_i(t)$ is modeled as a piecewise constant function (as is often the case), then it takes values $\alpha_i\in(0,B]$, $B<\infty$, belonging to a discrete set and $\bar{\alpha}_i$ obviously exists as the smallest positive value in this set. If we insist on modeling $\alpha_i(t)$ as a continuous or piecewise continuous process, then $\bar{\alpha}_i$ only exists for any $\alpha_i(t)$ that always changes from 0 to some arbitrarily small positive value $\bar{\alpha}_i$ through a jump (our analysis, however, is not affected by the possible presence of such chattering). Moreover, to avoid possible chattering around $x_i=\kappa_i$, we set $\beta_i(\boldsymbol{x}(t),\boldsymbol{\theta}) = \alpha_i(t)$ in Eq. (39) when $x_i(t)=\kappa_i$ and $0<\alpha_i(t)\leqslant C(t)\theta_i$, thus ensuring by Eq. (24) that the state remains at κ_i unless $\alpha_i(t)>C(t)\theta_i$.

In this specific model, the simple service capacity allocation mechanism adopted does not involve induced events. All exogenous and endogenous events are the same as those defined in the general setting of Sect. 5.1, except for endogenous events occurring as a result of the switching function $g(\boldsymbol{x}(t),\boldsymbol{\theta}) = x_i-\kappa_i$. For simplicity, we will refer to such events as "κ events" in the following discussion.

Using Eq. (11), we can derive the event time derivatives for κ events as follows:

$$\tau'_{k,j} = \frac{-x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - \beta_i(\boldsymbol{x}(\tau_k^-),\boldsymbol{\theta})}. \tag{41}$$

Clearly, the value of $\beta_i(\boldsymbol{x}(\tau_k^-),\boldsymbol{\theta})$ above depends on whether $x_i(\tau_k^-)<\kappa_i$ or $x_i(\tau_k^-)>\kappa_i$. For convenience, we define two separate events: an event denoted by ρ to represent the case $x_i(\tau_k^-)<\kappa_i$ and an event denoted by π to represent the case $x_i(\tau_k^-)>\kappa_i$. Let us examine next the IPA derivatives for the state when each of these two events takes place.

If a ρ event occurs, there are two possible cases:

Case 1 $0<\alpha_i(\tau_k)\leqslant C(\tau_k)\theta_i$. Based on Eq. (39), $x_i(\tau_k^+) = \kappa_i$ and the state remains unchanged until an exogenous event happens that changes the sign of $\alpha_i(t)-C(t)\theta_i$ causing the state to increase or decrease away from $x_i(t)=\kappa_i$. In this case, it follows from Eqs. (32) and (41) that

$$\begin{aligned} x'_{i,j}(\tau_k^+) &= x'_{i,j}(\tau_k^-) + \left[\alpha_i(\tau_k^-) - \beta_i(\boldsymbol{x}(\tau_k^-),\boldsymbol{\theta}) - 0\right] \\ &\quad\cdot\frac{-x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - \beta_i(\boldsymbol{x}(\tau_k^-),\boldsymbol{\theta})} \\ &= 0. \end{aligned} \tag{42}$$

Case 2 $\alpha_i(\tau_k) > C(\tau_k)\theta_i$. Based on Eq. (39), the state will cross κ_i and $\beta_i(\boldsymbol{x}(\tau_k^+),\boldsymbol{\theta}) = c_i(\boldsymbol{x}(\tau_k^+),\boldsymbol{\theta}) = C(\tau_k)\theta_i$, hence from Eqs. (32) and (41):

$$\begin{aligned} x'_{i,j}(\tau_k^+) &= x'_{i,j}(\tau_k^-) + \left[c_i(\boldsymbol{x}(\tau_k^+),\boldsymbol{\theta}) - \beta_{i,\min}\right] \\ &\quad\cdot\frac{-x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - \beta_{i,\min}} \\ &= A_{i,k}x'_{i,j}(\tau_k^-), \end{aligned} \tag{43}$$

where we have defined

$$A_{i,k} = \frac{\alpha_i(\tau_k^-) - C(\tau_k)\theta_i}{\alpha_i(\tau_k^-) - \beta_{i,\min}}.$$

If a π event occurs, there are also two possible cases:

Case 1 $0 < \alpha_i(\tau_k) \leqslant C(\tau_k)\theta_i$. This is similar to *Case 1* when a ρ event occurs and we get $x'_{i,j}(\tau_k^+) = 0$.

Case 2 $\alpha_i(\tau_k) = 0$. Based on Eq. (39), the state will cross κ_i and $\beta_i(\boldsymbol{x}(\tau_k^+), \boldsymbol{\theta}) = \beta_{i,\min}$ and from Eqs. (32) and (41):

$$\begin{aligned} x'_{i,j}(\tau_k^+) &= x'_{i,j}(\tau_k^-) + [\beta_{i,\min} - C(\tau_k)\theta_i] \\ &\quad \cdot \frac{-x'_{i,j}(\tau_k^-)}{\alpha_i(\tau_k^-) - C(\tau_k)\theta_i} \\ &= \frac{1}{A_{i,k}} x'_{i,j}(\tau_k^-). \end{aligned} \tag{44}$$

Note that in this case, $\alpha_i(\tau_k^-) = 0$, so that $\frac{1}{A_{i,k}} = \frac{\beta_{i,\min}}{c_i(\boldsymbol{x}(\tau_k),\boldsymbol{\theta})}$. However, we still write $1/A_{i,k}$ for ease of notation.

Between any two consecutive events, based on the specific definitions of $\beta_i(\boldsymbol{x}(t), \boldsymbol{\theta})$ in Eq. (39), we obtain from Eq. (35) for all $t \in [\tau_k, \tau_{k+1})$:

$$x'_{i,j}(t) = \begin{cases} x'_{i,j}(\tau_k^+) - C(\tau_k)(t-\tau_k), & x_{i,j}(t) > \kappa_i,\ j = i, \\ x'_{i,j}(\tau_k^+), & \text{otherwise}, \end{cases} \tag{45}$$

where the boundary condition $x'_{i,j}(\tau_k^+)$ is given by Eqs. (31), (42), (43) or (44) depending on the last event to occur. Applying Eq. (45) to an interval $[\tau_{k-1}, \tau_k)$, we can obtain $x'_{i,j}(\tau_k^-)$ as

$$x'_{i,j}(\tau_k^-) = \begin{cases} x'_{i,j}(\tau_{k-1}^+) - C(\tau_{k-1})(\tau_k - \tau_{k-1}), \\ \qquad\qquad x_{i,j}(t) > \kappa_i,\ j = i, \\ x'_{i,j}(\tau_{k-1}^+), \quad \text{otherwise}. \end{cases}$$

Therefore, Eqs. (43) and (44) become

$$\begin{aligned} x'_{i,j}(\tau_k^+) &= A_{i,k} x'_{i,j}(\tau_{k-1}^+), \\ x'_{i,j}(\tau_k^+) &= \frac{1}{A_{i,k}} \left(x'_{i,j}(\tau_{k-1}^+) - C(\tau_{k-1})(\tau_k - \tau_{k-1}) \right). \end{aligned}$$

We can now summarize the evolution of $x'_{i,j}(\tau_k^+)$ based *only* on $x'_{i,j}(\tau_{k-1}^+)$, i.e., the value of the state derivative at the previous event, along with the event time information τ_k, τ_{k-1} and $\alpha_i(\tau_k^-)$ required to evaluate $A_{i,k}$:

$$x'_{i,j}(\tau_k^+) = \begin{cases} A_{i,k} x'_{i,j}(\tau_{k-1}^+), & \rho \text{ occurs},\ \alpha_i(\tau_k) > C(\tau_k)\theta_i, \\ \frac{1}{A_{i,k}} x'_{i,j}(\tau_{k-1}^+) - \mathbf{1}[i=j] C(\tau_{k-1})(\tau_k - \tau_{k-1}), & \\ \qquad\qquad \pi \text{ occurs},\ \alpha_i(\tau_k) = 0, & \\ 0, \quad \text{otherwise}. \end{cases} \tag{46}$$

It is worthwhile noting that if $x'_{i,j}(0) = 0$, then $x'_{i,j}(t) = 0$ until a π event occurs at $t = \tau_k$ and $\alpha_i(\tau_k) = 0$ at which point $x'_{i,j}(\tau_k^+) = -C(\tau_{k-1})(\tau_k - \tau_{k-1})/A_{i,k}$.

Turning our attention to the cost derivatives, we adopt the cost functions defined in Eq. (27). Then, using the fact that $c_i(\boldsymbol{x}(t), \boldsymbol{\theta}, C(t)) = C(t)\theta_i$, the IPA derivative $\partial \mathcal{L}_i / \partial \theta_j$ in Eq. (36) becomes

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \theta_j} = \frac{1}{T} \Bigg[& \sum_{k=0}^{N_T} \mathbf{1}[i=j] p_i C(\tau_k) + \sum_{k \notin \Phi_i} \int_{\tau_k}^{\tau_{k+1}} x'_{i,j}(t)\, \mathrm{d}t \\ & + R_i \sum_{k \in \Phi_i} (\tau'_{k+1,j} - \tau'_{k,j}) \Bigg], \end{aligned}$$

where Φ_i is the set of all empty period intervals defined in Eq. (33). Using Eqs. (45) and (46), this can be further reduced to

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \theta_j} = & \frac{1}{T} \sum_{k \in \Psi_i} x'_{i,j}(\tau_k^+)(\tau_{k+1} - \tau_k) \\ & + \frac{1}{T} \sum_{k \in \Theta_i} \Big[x'_{i,j}(\tau_k^+)(\tau_{k+1} - \tau_k) \\ & - \mathbf{1}[i=j] \frac{C(\tau_k)}{2} (\tau_{k+1} - \tau_k)^2 \Big] \\ & + \frac{R_i}{T} \sum_{k \in \Phi_i} (\tau'_{k+1,j} - \tau'_{k,j}) \\ & + \frac{1}{T} \sum_{k=0}^{N_T} \mathbf{1}[i=j] p_i C(\tau_k), \end{aligned} \tag{47}$$

where Ψ_i and Θ_i are defined as

$$\Psi_i = \{k,\ \text{s.t. } 0 < x_i(t) \leqslant \kappa_i \text{ for all } t \in [\tau_k, \tau_{k+1})\}, \tag{48}$$

$$\Theta_i = \{k,\ \text{s.t. } x_i(t) > \kappa_i \text{ for all } t \in [\tau_k, \tau_{k+1})\}. \tag{49}$$

Note that the IPA estimators in Eq. (47) only depend on information at events times, which is easily obtained and computed using Eqs. (30), (31), (41) and (46).

5.2.1 Numerical example

In this section, we provide numerical results obtained on a two-class example of the game defined above. The inflow rates are set to be simple Markov modulated ON/OFF renewal processes, such that α_1 alternates between 0 and 2.5 and times between changes are exponentially distributed with mean 100 s, and α_2 alternates between 0 and 2 with interchanging times also exponentially distributed with mean 150 s. The total capacity $C(t)$ remains constant, i.e., $C(t) = 5$ for all t. We set $\kappa_1 = 5$, $\kappa_2 = 6$, $\beta_{1,\min} = 0.1$, $\beta_{2,\min} = 0.15$. We also assume equal unit prices and weights, i.e., $p_1 = p_2 = 20$, and $w_1 = w_2 = 1$.

With the IPA gradient estimators obtained in Eq. (47), we use the stochastic approximation algorithm given in Eq. (4) to carry out both system-centric and user-centric optimization, as in Refs. [16,20]. In the system-centric approach, we minimize the overall cost

function in Eq. (28) and determine the optimal control (θ_1^s, θ_2^s). In the user-centric approach, each class alternates minimizing its own cost function J_i, by adjusting only its control parameter θ_i. Thus, at the kth iteration, class 1 "plays" by optimizing J_1 using $\partial \mathcal{L}_1/\partial \theta_1$ in Eq. (47), and obtains the optimal control $\theta_{1,k}$; hence the control of class 2 is set to $\theta_{2,k} = 1 - \theta_{1,k}$. Next, at the $(k+1)$th iteration, class 2 "plays" with starting point $(\theta_{1,k}, \theta_{2,k})$ and using $\partial \mathcal{L}_2/\partial \theta_2$ in Eq. (47), and obtains optimal control $\theta_{2,k+1}$; hence, $\theta_{2,k} = 1 - \theta_{2,k+1}$. In Fig. 2, we show optimization results using the IPA gradient estimators in Eq. (47) for two game realizations with different initial conditions. Also shown is the "actual" system-centric cost function obtained through exhaustive simulation. We can see that our algorithm leads to the "actual" optimal point which is approximately $(0.57, 0.43)$. However, the user-centric optimization will not converge to the actual optimal point, and different starting points will lead to different results as seen in the figure. In addition, in the user-centric optimization, the controls exhibit an oscillatory behavior as shown in Fig. 3. One way to avoid this oscillatory behavior is to adopt a negotiation scheme as proposed in Ref. [20]. In this scheme, consider the kth iteration and let the current allocation be $(\theta_{1,k}, \theta_{2,k})$. Then, class 1 evaluates its next "candidate" control, denoted by $\theta_{1,k}^1$ through Eq. (4) using the IPA estimate of $\partial \mathcal{L}_1/\partial \theta_1$ from Eq. (47), and hence the control for class 2 as well, denoted by $\theta_{2,k}^1$. Similarly, class 2 evaluates $\theta_{1,k}^2$ and $\theta_{2,k}^2$. Then, the new allocation for the next iteration is obtained from

$$\theta_{1,k+1} = \sum_{i=1}^{2} \zeta_i \theta_{1,k}^i, \quad \theta_{2,k+1} = \sum_{i=1}^{2} \zeta_i \theta_{2,k}^i, \tag{50}$$

where $\sum_{i=1}^{2} \zeta_i = 1$ and ζ_i is the "negotiation weight" for class i (agreed upon in advance), which reflects the class's relative "power" or a "price" that class i is willing to pay to influence the ultimate allocation. An attractive feature of this scheme is that, when the negotiation weights and step sizes for each class satisfy the sufficient condition in the following lemma, the user-centric optimization will also converge to the actual optimal point.

Lemma 5.1 Let $\{\eta_{i,n}\}$ be the sequence of step sizes adopted by class i in the user-centric optimization. If $\zeta_1 \eta_{1,n} = \zeta_2 \eta_{2,n}$ for all $n \in \{1, 2, \ldots\}$, then user-centric optimization will converge to (θ_1^s, θ_2^s).

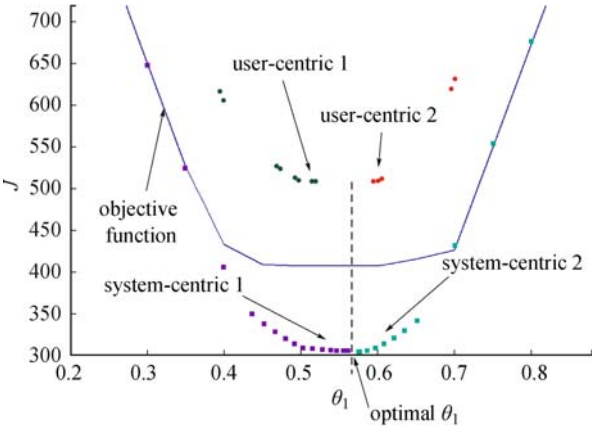

Fig. 2 Optimization results

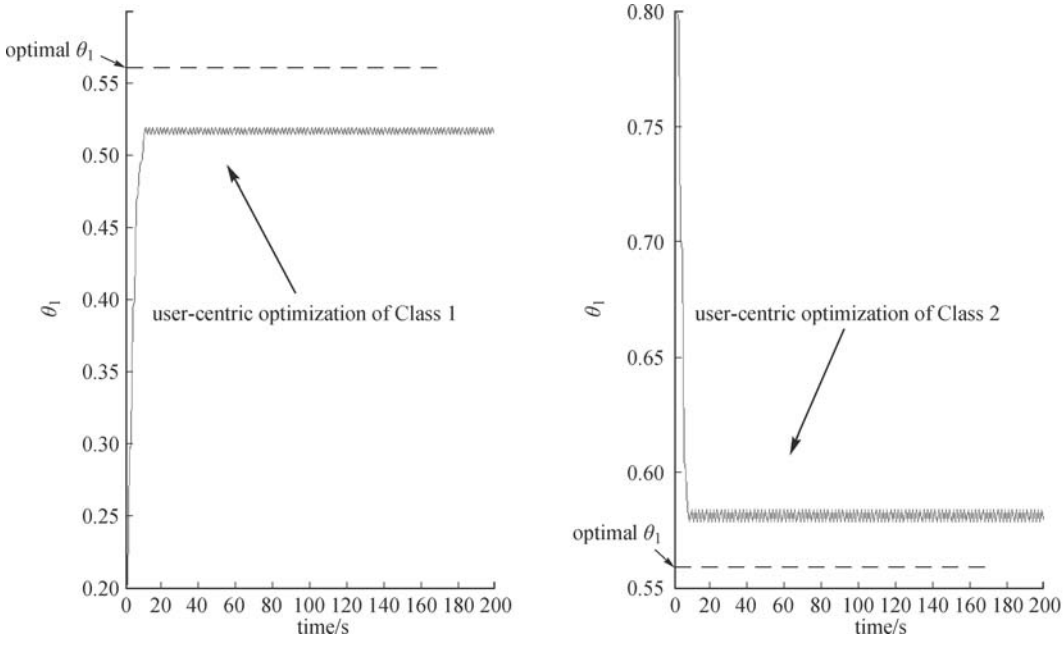

Fig. 3 Oscillatory behavior of control parameters

Proof Based on Eq. (50), for any iteration k,

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

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

$$\begin{aligned}\theta_{1,k+1} &= \theta_{1,k} - \eta_k\frac{\partial J_1(\theta)}{\partial\theta_1} - \eta_k\frac{\partial J_2(\theta)}{\partial\theta_1} \\ &= \theta_{1,k} - \eta_k\left(\frac{\partial J_1(\theta)}{\partial\theta_1} + \frac{\partial J_2(\theta)}{\partial\theta_1}\right) \\ &= \theta_{1,k} - \eta_k\frac{\partial J(\theta)}{\partial\theta_1}. \qquad (51)\end{aligned}$$

Similarly,

$$\theta_{2,k+1} = \theta_{2,k} - \eta_k\frac{\partial J(\theta)}{\partial\theta_2}. \qquad (52)$$

From Eqs. (51) and (52), 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. ■

We note that the above sufficient condition is quite restrictive. Generally, even if the oscillatory behavior is eliminated, there is still no guarantee that user-centric optimization will converge to (θ_1^s, θ_2^s).

6 Conclusions and future work

We have provided an overview of a recently developed general framework for IPA in SHSs and established some conditions under which IPA is particularly simple and efficient. When our goal is to develop an SHS as an abstraction of a complex DES, we have presented a systematic method for generating such abstractions. The proposed IPA framework opens up a new spectrum of applications where IPA may be used to study a class of stochastic non-cooperative games termed "resource contention games". In such games, multiple users compete for a sharable resource and IPA enables gradient-based optimization for both a system-centric and a user-centric perspective. We have provided numerical results for a simple version of a resource contention game to illustrate the effectiveness of the IPA framework and to contrast system-centric and user-centric optimization. In general, the corresponding solutions do not coincide, giving rise to what is referred to as "the price of anarchy". However, for at least one class of resource contention problems it was recently shown that these two solutions do in fact coincide [22], opening up an interesting directions at exploring conditions under which this is possible.

Acknowledgements This work 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.

References

1. Cassandras C G, Lygeros J. Stochastic Hybrid Systems. London: Taylor and Francis Group/CRC Press, 2006
2. Cassandras C G, Lafortune S. Introduction to Discrete Event Systems. 2nd ed. New York, NY: Springer, 2008
3. Ho Y C, Cao X R. Perturbation Analysis of Discrete Event Dynamic Systems. Boston, MA: Kluwer Academic Publisher, 1991
4. Glasserman P. Gradient Estimation via Perturbation Analysis. Boston, MA: Kluwer Academic Publisher, 1991
5. Cassandras C G, Wardi Y, Melamed B, Sun G, Panayiotou C G. Perturbation analysis for on-line control and optimization of stochastic fluid models. IEEE Transactions on Automatic Control, 2002, 47(8): 1234 – 1248
6. Anick D, Mitra D, Sondhi M M. Stochastic theory of a data-handling system with multiple sources. The Bell System Technical Journal, 1982, 61(8): 1871 – 1894
7. Liu B, Guo Y, Kurose J, Towsley D, Gong W B. Fluid simulation of large scale networks: Issues and tradeoffs. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications. 1999, 2136 – 2142
8. Connor D, Feigin G, Yao D D. Scheduling semiconductor lines using a fluid network model. IEEE Transactions on Robotics and Automation, 1994, 10(2): 88 – 98
9. Yu H, Cassandras C G. Perturbation analysis of feedback-controlled stochastic flow systems. IEEE Transactions on Automatic Control, 2004, 49(8): 1317 – 1332
10. Wardi Y, Adams R, Melamed B. A unified approach to infinitesimal perturbation analysis in stochastic flow models: The single-stage case. IEEE Transactions on Automatic Control, 2009, 55(1): 89 – 103
11. Sun G, Cassandras C G, Panayiotou C G. Perturbation analysis and optimization of stochastic flow networks. IEEE Transactions on Automatic Control, 2004, 49(12): 2113 – 2128
12. Yu H, Cassandras C G. Perturbation analysis and feedback control of communication networks using stochastic hybrid

models. Journal of Nonlinear Analysis, 2006, 65(6): 1251 – 1280

13. Cassandras C G, Wardi Y, Panayiotou C G, Yao C. Perturbation analysis and optimization of stochastic hybrid systems. European Journal of Control, 2010, 16(6): 642 – 664
14. Yao C, Cassandras C G. Perturbation analysis of stochastic hybrid systems and applications to some non-cooperative games. In: Proceedings of the 10th International Workshop on Discrete Event Systems. 2010, 69 – 74
15. Kushner H J, Yin G G. Stochastic Approximation Algorithms and Applications. New York, NY: Springer-Verlag, 1997
16. Yao C, Cassandras C G. Perturbation analysis and optimization of multiclass multi-objective stochastic flow models. In: Proceedings of the 48th IEEE Conference of Decision and Control. 2009, 914 – 919
17. Rubinstein R. Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks. New York, NY: John Wiley and Sons, 1986
18. Sun G, Cassandras C G, Panayiotou C G. Perturbation analysis of multiclass stochastic fluid models. Journal of Discrete Event Dynamic Systems: Theory and Applications, 2004, 14(3): 267 – 307
19. Chen M, Hu J Q, Fu M C. Perturbation analysis of a dynamic priority call center. IEEE Transactions on Automatic Control, 2008, 55(5): 1191 – 1196
20. Yao C, Cassandras C G. Perturbation analysis and resource contention games in multiclass stochastic fluid models. In: Proceedings of the 3rd IFAC Conference on Analysis and Design of Hybrid Systems. 2009, 256 – 261
21. Yao C, Cassandras C G. Perturbation analysis and resource contention games in multiclass stochastic fluid models. Nonlinear Analysis: Hybrid Systems (in press)
22. Yao C, Cassandras C G. A solution of the lot sizing problem as a stochastic resource contention game. In: Proceedings of the 49th IEEE Conference of Decision and Control. 2010, 6728 – 6733

Chen YAO is a final year Ph.D candidate in the Division of Systems Engineering at Boston University. He received a Bachelor degree in automatic control from Zhejiang University in 2006 and a Master degree in systems engineering from Boston University in 2009. He has been working as a Research Assistant in the Center for Information and Systems Engineering (CISE) at Boston University since 2006. His research interests lie in the areas of discrete event and hybrid systems, stochastic optimization, and cooperative control, with applications to manufacturing systems, communication systems, and robotics. He is the recipient of several awards, including the General Chair's Recognition Award of Interactive Sessions in the 48th IEEE Conference of Decision and Control (CDC), and Dean's Fellowship at Boston University in 2006. He is a student member of IEEE and INFORMS.

Christos G. CASSANDRAS is Head of the Division of Systems Engineering and Professor of Electrical and Computer Engineering at Boston University. He is also co-founder of Boston University's Center for Information and Systems Engineering (CISE). He received degrees from Yale University (B.S., 1977), Stanford University (M.S.E.E., 1978), and Harvard University (S.M., 1979; Ph.D, 1982). In 1982 – 1984 he was with ITP Boston, Inc., where he worked on the design of automated manufacturing systems. In 1984 – 1996 he was a faculty member at the Department of Electrical and Computer Engineering, University of Massachusetts/Amherst. He specializes in the areas of discrete event and hybrid systems, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published over 300 refereed papers in these areas, and five books. He has guest-edited several technical journal issues and serves on several journal Editorial Boards. He has recently collaborated with The MathWorks, Inc. in the development of the discrete event and hybrid system simulator SimEvents. Dr. Cassandras was Editor-in-Chief of the IEEE Transactions on Automatic Control from 1998 through 2009 and has also served as Editor for Technical Notes and Correspondence and Associate Editor. He is the 2011 President-Elect of the IEEE Control Systems Society (CSS) and has served as Vice President for Publications and on the Board of Governors of the CSS. He has chaired the CSS Technical Committee on Control Theory, and served as Chair of several conferences. He has been a plenary speaker at various international conferences, including the American Control Conference in 2001 and the IEEE Conference on Decision and Control in 2002. He is the recipient of several awards, including the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for Discrete Event Systems: Modeling and Performance Analysis, and a 1991 Lilly Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the IEEE and a Fellow of the IFAC.