

A Solution to the Optimal Lot Sizing Problem as a Stochastic Resource Contention Game

Chen Yao, Christos G. Cassandras, *Fellow, IEEE*

Abstract—We present a new way to solve the "lot sizing" problem viewed as a stochastic non-cooperative resource contention game. We develop a Stochastic Flow Model (SFM) for polling systems with non-negligible changeover times enabling us to formulate lot sizing as an optimization problem without imposing constraints on the distributional characteristics of the random processes in the system. Using Infinitesimal Perturbation Analysis (IPA) methods we derive gradient estimators of the performance metrics of interests with respect to the lot size parameters and prove they are unbiased. We then derive an on-line gradient-based algorithm for obtaining optimal lot sizes from both a system-centric and user-centric perspective. Uncharacteristically for such cases, there is no gap between the two solutions in the two-class case for which we have obtained explicit numerical results. We derive a proof of this phenomenon for a deterministic version of the problem, suggesting that lot-sizing-like scheduling policies in resource contention problems have a natural property of balancing certain user-centric and system-centric performance metrics.

Note to Practitioners—This paper is motivated by the "lot sizing" problem in manufacturing which involves the determination of the optimal number of parts combined to form a "lot" for each of several part types differing in their processing times, raw material supplying rates, etc. Using better selected lot sizes decreases the average lead time, and ultimately leads to larger throughput. In addition, lot sizing provides an inexpensive way to improve performance by controlling a simple parameter, especially when compared to complicated manufacturing re-engineering processes. The paper proposes an algorithm to calculate optimal lot sizes for multiple types of manufacturing parts. This algorithm only relies on data that are either readily observable or easily calculated from operating production systems. It can be easily programmed and used for on-line estimation of optimal lot sizes. Further, the paper examines the problem from the point of view of a central coordinator aiming to optimize a system-wide objective and can select all lot sizes or as a "game" where each part type controller can individually select its own lot size to optimize its own objective. It is shown that both approaches lead to the same solution. This property is attractive since it indicates an inherent fairness in treating different part types and suggests that similar mechanisms may be used in other applications with similar resource contention features.

Index Terms—Stochastic Flow Model, Lot-Sizing, Perturbation Analysis

I. INTRODUCTION

A large class of Discrete Event Systems (DES) involves the control of resources allocated to users, ultimately aiming to optimize various performance metrics. In this paper, we consider a family of such problems in which a single resource can "cycle" through multiple users and provide service to each

one according to some policy that defines the amount of time or number of tasks devoted to that user before it moves to a different one. The problem is complicated by the fact that the resource cannot instantaneously switch between users and doing so incurs a delay referred to as "changeover time" or "setup time." This models, for example, transportation systems where the resource physically moves from one location to another to pick up or deliver users (passengers or goods). In what follows, we first review the previous relevant work, and then introduce our proposed solution.

A. Previous Work

Problems as described above have been extensively studied in the literature as *limited service polling models* [1], where each user is represented as a queue of tasks waiting for service. In these models, each queue receives service until K tasks are processed (known as *K-limited service*) or until the time spent in the current queue exceeds a certain threshold T (known as *time-limited service*). The analysis carried out aims at deriving or estimating the average system time given the parameter K or T depending on the policy considered. Exact solutions can only be obtained for some special cases such as assuming symmetry and exponentially distributed service times as in [2], or two users only and no changeover time as in [3]. Alternatively, approximation methods may be used. For example, in [4], an upper bound was given for the average delay of a symmetric system with general service time distribution, by assuming that the system will keep serving the current queue till it's empty; in [5], authors studied the *time-limit service model*, using an approximation by assuming the time thresholds as exponentially distributed random variables, and then estimated the average delay using discrete Fourier transforms.

The *lot sizing* problem falls within the class of polling models and is mostly encountered in production planning for manufacturing systems. In this problem, changeover/setup times can be significant and a "lot" is defined as a set of tasks that must be performed by the resource in between successive changeovers. In the manufacturing system literature, the main concern is to determine lot sizes of manufactured parts for several future periods that minimize the sum of setup and inventory holding costs over a planning horizon, while satisfying a known demand in discrete time. The study of this problem has its origins in the Economic Order Quantity (EOQ) model proposed in [6], extended through the years ([7], [8], [9], [10]) and eventually formulated mathematically as a mixed integer programming problem with binary variables representing setups for each job-period combination. The

Manuscript submitted XXX. This work is supported in part by NSF under Grant EFRI-0735974, 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.

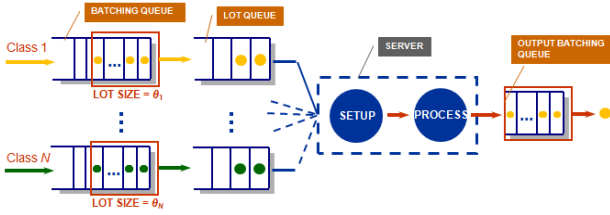


Fig. 1. Queueing-based model for the lot-sizing problem.

problem is NP-hard [11], so that various heuristics have been proposed as in [12],[13],[14],[15], [16], [17], [18], [19]. In addition, extensions to multi-stage lot-sizing problems have been considered in [20],[21],[22], [23], [24], where users (parts) go through a sequence of resources (machines) leading to formidable complexity even if one assumes unlimited resource capacities. This line of work is based on discrete-time models, usually assuming fixed setup costs and inventory holding costs, and it ignores random effects in the job arrival and service processes since there is actually no notion of “jobs” in such discrete-time models.

In view of the above limitations, research eventually shifted toward studying the lot-sizing problem based on queueing systems with the mean system time as a cost function instead of setup costs. In these models, a common resource (server) is scheduled to cycle through multiple users (queues, each containing jobs of the same “class”) and arriving jobs of each class are batched on a First-Come-First-Served (FCFS) basis to define a “lot” comprised of θ_i jobs. The objective then is to select proper lot sizes for different classes in order to minimize the overall mean system time. Figure 1 illustrates this model with N different job classes representing different part types. The interarrival and processing times for jobs of class i are all random variables and vary among different classes. Arriving jobs of class i wait until θ_i of them can be batched on a FCFS basis to define a single lot (note that θ_i need not be integer-valued, since the actual lot may be chosen to be the floor $\lfloor \theta_i \rfloor$). Once there is a lot waiting in the i th queue, the server can start processing it according to some given schedule. We will assume this schedule is based on a *round-robin* scheme, i.e., the classes have been ordered in advance. When the server switches to class i , a setup time S_i is incurred, which is generally also a random variable. When a lot is served, jobs are processed individually and must wait in an output queue until the entire lot is complete. It is only at that time that the lot is released to the next server (or leaves the system). Clearly, during a setup, the server is idle, so it is desirable to minimize the total setup time. If lot sizes are small, the server engages in frequent setups and utilization is low; on the other hand, if lots are large, then classes experience long queueing delays in both the input queue (until the lot is formed) and output queue (until all jobs in the lot are processed) shown in Figure 1. Therefore, there is a clear tradeoff in selecting lot sizes and, generally, an optimal value depending on how the tradeoff is mathematically expressed.

Previous work on this model has focused on deriving relationships between the mean system time and lot sizes. Closed-

form results can only be obtained for a simple Markovian queueing model in [25] or a single $M/G/1$ queue in [26]. For multiclass systems, one can only resort to approximation methods to capture this relationship as in [27],[28]. For example, in [28], the system is approximated by an $M/G/1$ queue with the service time distribution given by the mix of processing times across batches of all classes. There is no guarantee on the quality of these approximation methods which are mainly used for qualitative analysis and gaining insights, rather than aiming to formally control and optimize the system. An alternative approach based on estimating performance sensitivities and using a “surrogate” optimization process was also proposed in [29]. It is worth noting that lot sizing provides an opportunity to optimize system performance by controlling simple parameters (lot sizes) and is thus an inexpensive alternative to complex manufacturing re-engineering processes. Moreover, by periodically adjusting the parameter values one can react to random effects in the environment. The problem is also interesting in that it may be viewed from the perspective of either the system trying to optimize the overall average system time while maintaining high server utilization or the individual user (the i th class) trying to optimize its own average system time. This gives rise to a resource contention game which often results in a gap between the system-centric and user-centric optimal solutions. This gap is commonly referred to as the “price of anarchy.”

B. SFM and IPA Method

In this paper, we propose a new way to tackle the lot-sizing problem, by first constructing a *Stochastic Flow Model* (SFM) for the problem. SFMs, introduced in [30], form a class of stochastic hybrid systems which can be used as abstractions of DES whose event-driven dynamics are too complex to analyze. The time-driven component of a SFM captures general-purpose flow dynamics and the event-driven component describes switches, controlled or uncontrolled, that alter the flow dynamics. Fluid models have been used to analyze or efficiently simulate various settings where users compete over different sharable resources (e.g., [31], [32], [33], [34], [35], [36]). Unlike traditional fluid models where the flow rates involved are treated as fixed parameters, a SFM has the extra feature of treating flow rates as stochastic processes. With only minor technical conditions imposed on the properties of such processes, it is therefore possible to model very general dynamics (e.g., [37].) Moreover, while recognizing that fluid models cannot always provide accurate representations for the purpose of analyzing the performance of the underlying DES, the purpose of a SFM in our case is to enable the estimation of performance sensitivity estimates through which we can achieve effective control and optimization. Thus, the value of a fluid model lies in capturing those system features needed to design an effective controller that can potentially optimize performance without any attempt at estimating the corresponding performance value with accuracy. Our approach is based on Infinitesimal Perturbation Analysis (IPA), which uses data from a single observed sample path to yield unbiased gradient estimates of performance metrics with respect to lot-sizing

parameters and then drives an on-line optimization algorithm to obtain optimal lot sizes, thus circumventing the difficulty of obtaining closed-form analytical solutions. In addition, the gradient estimators obtained through IPA are independent of the distributional information of the random processes involved. IPA was originally developed for DES [38] to obtain unbiased gradient estimates of performance metrics. 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. This difficulty has been shown to be circumvented in SFMs and IPA has been applied in optimizing some challenging systems that include blocking phenomena and a variety of feedback control mechanisms (e.g., [39], [40], [41], [42].) In addition, recent work on multiclass SFMs, for example, [43] and [44], has opened up the opportunity to study the difference between the usual *system-centric* optimization, where a system-wide metric is optimized, and a *user-centric* approach, where classes optimize their own performance function, thus leading to a non-cooperative resource contention game setting. As mentioned above, the two solutions are usually different. However, an interesting result of our analysis is that the system-centric and user-centric optimal solutions coincide. This suggests that the fundamental lot sizing mechanism in scheduling resources over non-cooperating users is characterized by this attractive property and may be applicable to broader problems of this type.

In Section II, we develop a SFM for the lot-sizing problem, classify all events that occur in the system, and give a formal representation of the system as a stochastic hybrid automaton. In Section III, we define the optimization problem and derive IPA gradient estimators based on the general framework developed in [37]. In Section IV, we prove the unbiasedness of the derived IPA estimators, and apply them to solve the lot sizing problem for the original DES. We also provide simulation results to illustrate our optimization algorithm and contrast the difference between system-centric and user-centric perspectives. Observing that there is no “price of anarchy” in this resource contention game, we prove this fact for the case of deterministic arrival, setup, and service processes and provide an explanation of this fact.

II. A STOCHASTIC FLOW MODEL (SFM) FOR THE LOT-SIZING PROBLEM

The SFM shown in Figure 2 is the counterpart of Figure 1. There are N classes of flows (jobs) entering the system with rates $\{r_i(t)\}$, $i = 1, \dots, N$, and sharing a single resource (server). Each flow has its own queue, which is associated with a lot-sizing parameter $\theta_i \in \mathbb{R}^+$, and $x_i(\theta, t) \in \mathbb{R}^+$ denotes the class i queue content at time t . We then define a vector $\mathbf{x}(\theta, t) = (x_1(\theta, t), \dots, x_i(\theta, t), \dots, x_N(\theta, t))'$, where $\theta = (\theta_1, \dots, \theta_i, \dots, \theta_N)'$. For notational simplicity, in the rest of the paper we will write $x_i(\theta, t)$ as $x_i(t)$ when no confusion arises. The server can only process one class at a time and switches among queues in round-robin fashion (modulo N). When it switches to queue i from the previous queue, a changeover time s_i is required before it can start processing

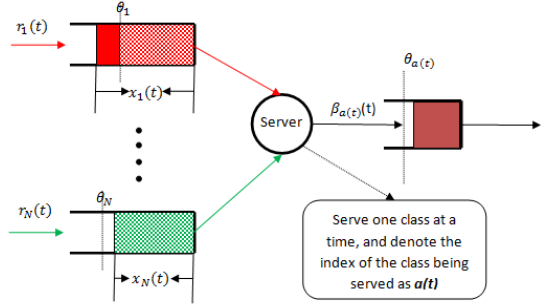


Fig. 2. SFM for Lot-sizing Problem

a flow from class i . If $x_i(t) \geq \theta_i$ at the time when the changeover is complete, then the server starts processing class i with rate $\beta_i(t)$; if $x_i(t) < \theta_i$, the server waits until $x_i(t) = \theta_i$. The index of the class the server is serving at time t is denoted by $a(t) \in \{1, \dots, N\}$. All processed flow enters an output queue where it remains until the content of the output queue reaches $\theta_{a(t)}$.

Associated with this stochastic hybrid system are several real-valued and non-negative random processes which are all defined on a common probability space (Ω, \mathcal{F}, P) . The arrival flow processes $\{r_i(t)\}$, $i = 1, \dots, N$, characterize the arrival rates of tasks at time t and the service flow processes $\{\beta_i(t)\}$, characterize their processing capacities. In addition, the changeover times s_i are random, with $\{s_i(k)\}$ defining random processes over the number of switches $k = 1, 2, \dots$ at each $i = 1, \dots, N$. For notational simplicity, we omit the k dependencies in $s_i(k)$ and write only s_i in what follows. We are interested in the behavior of this SFM over a finite time interval $[0, T]$. Regarding the arrival and service flow processes, we will impose no restrictions on them as far as the probability laws that characterize them are concerned, but will make the following assumption:

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

In addition to $a(t)$, $\mathbf{x}(t)$ defined above, let us define $y(\theta, t) \in \mathbb{R}^+$ to be the content of the output queue and $z(t) \in \mathbb{R}^+$ to be a “clock” value that represents the time elapsed since last switch from one queue to the next. As in the case of $x_i(\theta, t)$, we will write $y(t)$ instead of $y(\theta, t)$ when no confusion arises. Then, the dynamics of each state variable $x_i(t)$, $i = 1, \dots, N$, and of the output queue content $y(t)$ are given by

$$\frac{dx_i(t)}{dt^+} = \begin{cases} r_i(t) - \beta_i(t) & \text{if } a(t) = i, z(t) \geq s_{a(t)} \\ & \text{and } x_i(t) + y(t) \geq \theta_i \\ r_i(t) & \text{otherwise} \end{cases} \quad (1)$$

$$\frac{dy(t)}{dt^+} = \begin{cases} \beta_{a(t)}(t) & \text{if } z(t) \geq s_{a(t)}, y(t) < \theta_{a(t)} \\ & \text{and } x_{a(t)}(t) + y(t) \geq \theta_{a(t)} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$y(t) = 0 \quad \text{if } y(t^-) = \theta_{a(t^-)} \quad (3)$$

In (1), the condition $x_i(t) + y(t) \geq \theta_i$ indicates that there is

sufficient flow from class i to form a lot; if $y(t) = 0$ and $x_i(t) < \theta_i$, then $\frac{dx_i(t)}{dt^+} = r_i(t)$ and $\frac{dy(t)}{dt^+} = 0$, which implies the system remains idle until class i accumulates enough flow to form a lot. The condition $z(t) \geq s_{a(t)}$ simply indicates that the changeover is complete and the server may process a lot. In (2), the output queue fills up with rate $\beta_{a(t)}(t)$ as long as a lot is formed and the changeover is complete at the currently selected queue. The condition (3) is based on the fact that when $y(t^-) = \theta_{a(t^-)}$, the server has finished serving a full lot of the current class $a(t)$ and is ready to switch to the next class; when this happens, all flow in the output queue leaves the system thus resetting the output queue to an empty state $y(t) = 0$. Along the same lines, the dynamics of the clock variable $z(t)$ are

$$\begin{aligned} \frac{dz(t)}{dt^+} &= 1 \quad \text{if } y(t) < \theta_{a(t)} \\ z(t) &= 0 \quad \text{if } y(t^-) = \theta_{a(t^-)} \end{aligned} \quad (4)$$

where (4) indicates that the clock is also reset when the server switches to a new class.

A. Event Classification

Following the event classification for general stochastic hybrid systems in [37], there are three types of events that can occur in the SFM of Figure 2 and cause discrete state transitions:

1. Exogenous events. An event at time τ_k is *exogenous* if τ_k is independent of the controllable vector θ . In this SFM, exogenous events correspond to changes in the arrival flow rates $r_i(t)$ or the processing rates $\beta_i(t)$, $i = 1, \dots, N$.

2. Endogenous events. An event occurring at time τ_k is *endogenous* if there exists a continuously-differentiable function $g_k(x, y, \theta)$ (see also [37]), such that

$$\tau_k = \min \{t : t > \tau_{k-1}, g_k(\mathbf{x}(t), y(t), \theta) = 0\} \quad (5)$$

Based on the specific $g_k(x, y, \theta)$ of interest, we can further classify endogenous events as follows:

2.1. Switching events. These events occur when the server completes processing a lot of class $a(\tau_k^-)$ and switches to the next class, i.e., when $y(t)$ reaches $\theta_{a(t)}$ from below. Therefore,

$$g_k(\mathbf{x}, y, \theta) = y(t) - \theta_{a(t)} \quad (6)$$

2.2. Lot-forming events. These events occur when the current class accumulates sufficient flow to form a lot. If $z(\tau_k^-) < s_{a(\tau_k)}$, then from (1) the lot-forming event cannot initiate service and the system dynamics remain unchanged. Therefore, lot-forming events are only of interest when $z(\tau_k^-) \geq s_{a(\tau_k)}$ and $x(t)$ reaches $\theta_{a(t)}$ from below; in what follows, lot-forming events will only refer to events that satisfy these two conditions. Therefore,

$$g_k(\mathbf{x}, y, \theta) = x_{a(t)}(t) - \theta_{a(t)} \quad (7)$$

3. Induced events. An event occurring at time τ_k is *induced* if

$$z(\tau_k^-) = s_{a(\tau_k)} \quad (8)$$

Such an event occurs when a changeover is completed after switching to the current queue and is “induced” by the most recent switching event that initiates the associated changeover process. Letting τ_{k_s} denote the time of this switching event ($\tau_{k_s} < \tau_k$), we have

$$\tau_k = \tau_{k_s} + s_{a(\tau_k)} \quad (9)$$

Similar to a lot-forming event, an induced event is only of interest when $x_{a(\tau_k^-)}(\tau_k^-) + y(\tau_k^-) \geq \theta_{a(\tau_k^-)}$, and in what follows we will refer to induced events as only those that satisfy both this condition and (8). We note that these induced events are of a simpler form than those defined in [43] and [44], and are consistent with the general definitions given in [37] without requiring as much notation.

Observe that lot-forming events and induced events both initiate processing at the current queue and affect the dynamics of $y(t)$. Thus, in the following, we will sometimes refer to them as *service start events* for convenience.

B. Stochastic Hybrid Automaton Model

A stochastic hybrid automaton model for the SFM described thus far is shown in Figure 3, in which we limit ourselves to two classes for simplicity. There are eight discrete states in this model, represented by the circles in Figure 3, with associated time-driven dynamics inside the circles and guard and reset conditions shown on the directed arcs. States 1-4 represent the following four processes in the system: At state 1(4) the server is processing flow from class 1(2); at state 3(2) the server has switched to class 1(2) but is not yet ready to start processing. States 5-8 are different from 1-4 in that the system transitions into them with no actual change in the time-driven dynamics. They represent the two different ways that the server can start processing after a switching event. For example, at state 3 the server is switched to class 1 and must both complete a changeover time s_1 and have a flow amount of at least θ_1 before it can start serving, i.e., enter state 1. Thus, we have two possibilities: (i) when the changeover is complete, i.e., $z = s_1$, condition $x_1 < \theta_1$ holds, in which case state 5 is entered, and (ii) when $x_1 \geq \theta_1$ holds, but the changeover is still not complete, i.e., $z < s_1$, in which case state 6 is entered. States 5 and 6 have the same time-driven dynamics but different invariant conditions, and the system will transition to state 1 either by a lot-forming event from state 5 or by an induced event from state 6.

There are several possible variants of this model. For example, in one such variant a non-idling policy is adopted, i.e., the server immediately starts processing after a changeover is completed, regardless of whether a lot is fully formed or not. The stochastic hybrid automaton model for this variant is shown in Figure 4, where there are only four discrete states and no lot-forming events. The analysis developed for our current model can be readily modified to apply on this variant, and we will also include simulation results for this variant model to validate the effectiveness of our approach on it. Another variant (not considered here) arises when following a switching event to queue i the condition $x_i(t) = 0$ applies and the server immediately switches to the next queue. This

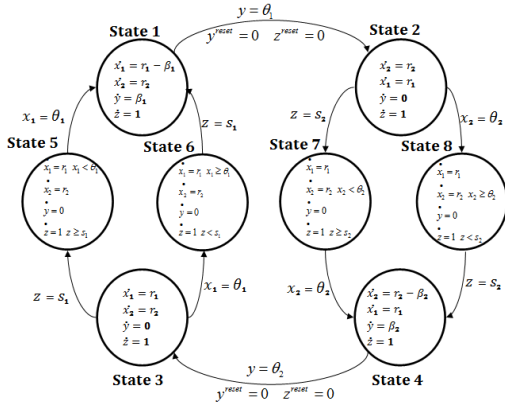


Fig. 3. The Stochastic Hybrid Automaton Model

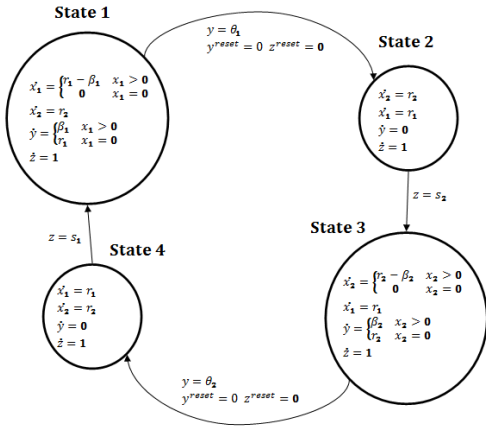


Fig. 4. The Stochastic Hybrid Automaton Model for the Variant Model with Non-Idling Policy

is normally less interesting because it corresponds to a lightly loaded system.

III. PERFORMANCE OPTIMIZATION

An optimization problem for our SFM is defined by viewing $\theta = (\theta_1, \dots, \theta_N)$ as a controllable parameter vector and seeking to optimize performance metrics of the form

$$J(\theta; x(0), T) = E[\mathcal{L}(\theta; x(0), T)] \quad (10)$$

where $\mathcal{L}(\theta; x(0), T)$ is a sample function of interest evaluated in the interval $[0, T]$ with initial conditions $\mathbf{x}(0)$. In the lot-sizing problem, we are typically interested in minimizing the mean system time for each class or equivalently the mean workload (i.e., queue contents) in the SFM. In this paper, we consider the sample performance function

$$\mathcal{L}(\theta; x(0), T) = \sum_{i=1}^N \gamma_i Q_i(\theta; x(0), T) \quad (11)$$

where $\gamma_i \in \mathbb{R}^+$, $i = 1, \dots, N$, are weight parameters which can be interpreted as unit holding costs of different classes, and

$$Q_i(\theta) = \frac{1}{T} \int_0^T [x_i(t, \theta) + \mathbf{1}(a(t) = i) \cdot y(t)] dt \quad (12)$$

where $\mathbf{1}(m = n)$ is the usual indicator function such that $\mathbf{1}(m = n) = 1$ if $m = n$ and 0 otherwise. Observe that $\mathbf{1}(a(t) = i) \cdot y(t)$ represents the delay experienced by jobs in the output queue of the actual DES while waiting for the remaining jobs of the lot under process.

Because of the multiclass nature of this problem, as mentioned in the introduction, in addition to the usual *system-centric* optimization focusing on the objective $J(\theta; x(0), T)$, each class (user) may solve its own optimization problem with a performance metric of the form $J_i(\theta) = E[Q_i(\theta)]$; this leads to a non-cooperative resource contention game that calls for solving N distinct *user-centric* optimization problems. We shall consider both in Section IV.

Since we do not wish to impose any limitations on the defining processes $\{r_i(t)\}$ and $\{\beta_i(t)\}$ (other than mild technical conditions), it is infeasible to obtain closed-form expressions for $J(\theta; x(0), T)$. Therefore, we resort to iterative methods such as stochastic approximation algorithms (e.g., [45]) which are driven by estimates of the cost function gradient with respect to the parameter vector of interest. Thus, we are interested in estimating $\partial J / \partial \theta_i$ based on sample path data, where a sample path of the system may be directly observed or obtained through simulation. We then seek to obtain θ^* minimizing $J(\theta; x(0), T)$ through an iterative scheme of the form

$$\theta_{i,n+1} = \theta_{i,n} - \eta_n H_{i,n}(\theta_n; x(0), T, \omega_n), \quad n = 0, 1, \dots \quad (13)$$

where $H_{i,n}(\theta_{i,n}; x(0), T, \omega_n)$ is an estimate of $\partial J / \partial \theta_i$ evaluated at $\theta = (\theta_{1,n}, \theta_{2,n}, \dots, \theta_{N,n})$ and based on information obtained from a sample path denoted by ω_n . For our purposes, we shall consider T to be a fixed time horizon and evaluate performance over $[0, T]$. To simplify the analysis that follows, we will assume that $x_i(0) = 0$, for all i (in practice, it is possible to avoid this issue as explained, for example, in [41]). In addition, we will omit in subsequent notation the initial condition, the observation interval T and the sample path ω_n unless it is necessary to stress such dependence.

In order to execute an algorithm such as (13), we need to estimate $H_{i,n}(\theta_{i,n})$, i.e., the derivative $\partial J / \partial \theta_i$. The IPA approach is based on using the sample derivative $\partial \mathcal{L} / \partial \theta_i$ as an estimate of $\partial J / \partial \theta_i$. The strength of the approach is that $\partial \mathcal{L} / \partial \theta_i$ 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 $\partial \mathcal{L} / \partial \theta_i$ is an *unbiased* estimate of $\partial J / \partial \theta_i$, a property that allows us to use (13) in obtaining θ^* . We will return to this issue and concentrate first on deriving IPA estimates.

It is clear from (11) and (12) that obtaining sample derivatives of $Q_i(\theta)$ requires the sample derivatives of the states $x_i(\theta, t)$, $y(\theta, t)$ and of the event times $\tau_k(\theta)$ where the explicit dependence on the parameter θ is included here for emphasis. This is precisely the task of IPA which we present in the next section.

A. IPA Estimation

To simplify notation in the sequel, we define the following for all state and event time sample derivatives, $i, j = 1, \dots, N$:

$$x'_{i,j}(t) \equiv \frac{\partial x_i(t)}{\partial \theta_j}, \quad y'_j(t) = \frac{\partial y(t)}{\partial \theta_j} \quad \tau'_{k,j} \equiv \frac{\partial \tau_k}{\partial \theta_j} \quad (14)$$

1) *State Perturbations*: Let us rewrite the flow dynamics (1) over an interval $[\tau_{k-1}, \tau_k)$ as $\frac{dx_i(t)}{dt} = f_{i,k}(t)$ where $f_{i,k}(t) = r_i(t) - \beta_i(t)$ or $f_{i,k}(t) = r_i(t)$. We obtain the dynamics of the sample derivative $x'_{i,j}(t)$ defined in (14) as follows for all $t \in [\tau_{k-1}, \tau_k)$ (see also [37]):

$$\frac{d}{dt} x'_{i,j}(t) = \sum_{l=1}^2 \frac{\partial f_{i,k}(t)}{\partial x_l} x'_{l,j}(t) + \frac{\partial f_{i,k}(t)}{\partial y} y'_j(t) + \frac{\partial f_{i,k}(t)}{\partial \theta_j} \quad (15)$$

and since $\frac{\partial f_{i,k}(t)}{\partial x_i} = \frac{\partial f_{i,k}(t)}{\partial y} = \frac{\partial f_{i,k}(t)}{\partial \theta_j} = 0$ over all $t \in [\tau_{k-1}, \tau_k)$,

$$\frac{d}{dt} x'_{i,j}(t) = 0 \quad (16)$$

that is, the IPA derivative $x'_{i,j}(t)$ remains fixed in between consecutive events. In addition, because of the continuity of the queue content $x_i(t)$, we have $x_i(\tau_k^+) = x_i(\tau_k^-)$. Taking derivatives with respect to θ_j , we have:

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

and since $\frac{\partial x_i(\tau_k^+)}{\partial t} = f_{i,k+1}(\tau_k^+)$, $\frac{\partial x_i(\tau_k^-)}{\partial t} = f_{i,k}(\tau_k^-)$,

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) + [f_{i,k}(\tau_k^-) - f_{i,k+1}(\tau_k^+)] \tau'_{k,j} \quad (17)$$

Note that $f_i(t)$ can be discontinuous in t at event times $t = \tau_k$, hence $x'_{i,j}(\tau_k^+)$, $x'_{i,j}(\tau_k^-)$ above are generally different. Combining (16) with (17) we get

$$x'_{i,j}(t) = x'_{i,j}(\tau_k^+), \quad t \in [\tau_k, \tau_{k+1}) \quad (18)$$

Thus, the queue content derivatives are piecewise constant, with jumps according to (17) at event times. It therefore suffices to use (17) to track them on an event by event basis.

For the state variables $y(t)$ with the dynamics (2) and (3), we get

$$y'_j(t) = y'_j(\tau_k^+), \quad t \in [\tau_k, \tau_{k+1}) \quad (19)$$

Since $y(t)$ is continuous at service start events, i.e., $y(\tau_k^+) = y(\tau_k^-)$, taking derivatives with respect to θ_j , we get

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

where at service start events, $\frac{\partial y(\tau_k^+)}{\partial t} = \beta_{a(\tau_k^+)}(\tau_k^+)$, $\frac{\partial y(\tau_k^-)}{\partial t} = 0$, so that

$$y'_j(\tau_k^+) = y'_j(\tau_k^-) - \beta_{a(\tau_k^+)}(\tau_k^+) \cdot \tau'_{k,j} \quad (20)$$

In addition, since $y(t)$ is reset at switching events by (3), obviously if a switching event occurs at τ_k ,

$$y'_j(\tau_k^+) = 0 \quad (21)$$

In summary, (17) and (20)-(21) fully describe the propagation of the state derivatives from one event to the next, provided we can also evaluate all event time derivatives $\tau'_{k,j}$, $k = 1, 2, \dots$, as described next.

2) *Event Time Derivatives*: Recalling the event classification in Section II-A, we consider event time perturbations for each event type.

1. Exogenous events. By definition, all such events are independent of θ , therefore:

$$\tau'_{k,j} = 0 \quad (22)$$

2. Endogenous events.

2.1 Switching events. In this case, from (6) we have $y(t) = \theta_{a(t)}$ and taking derivatives with respect to θ_j gives:

$$y'_j(\tau_k^-) + \frac{\partial y(\tau_k^-)}{\partial t} \cdot \tau'_{k,j} = \begin{cases} 1 & a(\tau_k^-) = j \\ 0 & a(\tau_k^-) \neq j \end{cases}$$

so that, using (2),

$$\tau'_{k,j} = \begin{cases} (1 - y'_j(\tau_k^-)) / \beta_{a(\tau_k^-)}(\tau_k^-) & a(\tau_k^-) = j \\ -y'_j(\tau_k^-) / \beta_{a(\tau_k^-)}(\tau_k^-) & a(\tau_k^-) \neq j \end{cases} \quad (23)$$

2.2 Lot-forming events. In this case, from (7) we have $x_{\alpha(t)}(t) = \theta_{\alpha(t)}$ and taking derivatives with respect to θ_j gives:

$$x'_{\alpha(\tau_k),j}(\tau_k^-) + \frac{\partial x_{\alpha(\tau_k)}(\tau_k^-)}{\partial t} \cdot \tau'_{k,j} = \begin{cases} 1 & a(\tau_k) = j \\ 0 & a(\tau_k) \neq j \end{cases}$$

and, using (1), we have

$$\tau'_{k,j} = \begin{cases} \frac{1 - x'_{\alpha(\tau_k),j}(\tau_k^-)}{r_{\alpha(\tau_k)}(\tau_k)} & a(\tau_k) = j \\ -\frac{x'_{\alpha(\tau_k),j}(\tau_k^-)}{r_{\alpha(\tau_k)}(\tau_k)} & a(\tau_k) \neq j \end{cases} \quad (24)$$

3. Induced events. By definition, we know such an event is triggered by a switching event at τ_{k_s} , therefore, taking derivatives with respect to θ_j on both sides of (9) gives $\tau'_{k,j} = \tau'_{k_s} + \partial s_{\alpha(\tau_k)} / \partial \theta_j$ and because changeover times s_i are independent of lot sizes, this reduces to

$$\tau'_{k,j} = \tau'_{k_s} \quad (25)$$

3) *IPA Derivative Estimation Process*: We can now combine the results from the previous two sections to provide a complete description of the IPA derivative estimation process on an event by event basis. Keeping in mind that the values of these derivatives are updated only at event times, we need only specify this update process at each event. We proceed again using our event classification in Section II.

1. Exogenous events. Based on (22), and in conjunction with (17), for all $i, j = 1, 2, \dots, N$:

$$\tau'_{k,j} = 0, \quad x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-), \quad y'_j(\tau_k^+) = y'_j(\tau_k^-) \quad (26)$$

2. Endogenous events.

2.1 Switching events. If a switching event occurs at τ_k with $a(\tau_k^-) = i$, by (1), $f_{i,k}(\tau_k^-) = r_i(\tau_k^-) - \beta_i(\tau_k^-)$ and $f_{i,k+1}(\tau_k^+) = r_i(\tau_k^+)$. Therefore,

$$f_{i,k+1}(\tau_k^+) = \begin{cases} f_{i,k}(\tau_k^-) + \beta_i(\tau_k^-) & \text{if } i = a(\tau_k^-) \\ f_{i,k}(\tau_k^-) & \text{otherwise} \end{cases}$$

and then using (17), (21) and (23), we get

$$x'_{i,j}(\tau_k^+) = \begin{cases} y'_j(\tau_k^-) - 1 & \text{if } i = j = a(\tau_k^-) \\ y'_j(\tau_k^-) & \text{if } i = a(\tau_k^-) \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$y'_j(\tau_k^+) = 0 \quad (27)$$

2.2 Lot-forming events. Based on (1) and recalling that a lot forming event is of interest only under the condition that $z(\tau_k^-) \geq s_{a(\tau_k)}$, if $a(\tau_k) = i$, then $f_{i,k}(\tau_k^-) = r_i(\tau_k)$ and $f_{i,k+1}(\tau_k^+) = r_i(\tau_k^+) - \beta_i(\tau_k)$. Therefore,

$$f_{i,k+1}(\tau_k^+) = \begin{cases} f_{i,k}(\tau_k^-) - \beta_i(\tau_k^-) & \text{if } i = a(\tau_k^-) \\ f_{i,k}(\tau_k^-) & \text{otherwise} \end{cases}$$

so from (17), (20) and (24), we get

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) + \begin{cases} \frac{\beta_i(\tau_k)(1-x'_{i,j}(\tau_k^-))}{r_i(\tau_k)} & i = a(\tau_k^-), j = a(\tau_k^-) \\ \frac{\beta_i(\tau_k) \cdot x'_{i,j}(\tau_k^-)}{r_i(\tau_k)} & i = a(\tau_k^-), j \neq a(\tau_k^-) \\ 0 & \text{otherwise} \end{cases}$$

$$y'_j(\tau_k^+) = y'_j(\tau_k^-) - \beta_{a(\tau_k)}(\tau_k) \cdot \tau'_{k,j} \quad (28)$$

3. Induced events. Similar to lot-forming events, using equations (17), (20) and (25), we have

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-) + \beta_i(\tau_k) \cdot \tau'_{k,j}, \quad i = a(\tau_k)$$

$$x'_{i,j}(\tau_k^+) = x'_{i,j}(\tau_k^-), \quad i \neq a(\tau_k) \quad (29)$$

$$y'_j(\tau_k^+) = y'_j(\tau_k^-) - \beta_{a(\tau_k)}(\tau_k) \cdot \tau'_{k,j}$$

Based on (26) through (29), we can evaluate all $x'_{i,j}(t)$ and $\tau'_{k,j}$ along a given sample path. We can then return to (12) and evaluate the performance metric derivatives $\frac{\partial Q_i(\theta)}{\partial \theta_j}$ as described next. Taking derivatives of $Q_i(\theta)$ with respect to θ_j , we get

$$\frac{\partial Q_i(\theta)}{\partial \theta_j} = \frac{1}{T} \sum_{k=1}^{N_T} [\tau'_{k,j} \cdot (x_i(\tau_k, \theta) + \mathbf{1}(a(\tau_k) = i) y(\tau_k)) + \int_{\tau_{k-1}}^{\tau_k} (x'_{i,j}(t, \theta) + \mathbf{1}(a(t) = i) y'_j(t)) dt - \tau'_{k-1,j} \cdot (x_i(\tau_{k-1}, \theta) + \mathbf{1}(a(\tau_{k-1}) = i) y(\tau_{k-1}))]$$

where N_T is the total number of events occurring in $[0, T]$. Based on (19) and (18), we can reduce this to

$$\frac{\partial Q_i(\theta)}{\partial \theta_j} = \frac{1}{T} \sum_{k=1}^{N_T} [\tau'_{k,j} \cdot (x_i(\tau_k, \theta) + \mathbf{1}(a(\tau_k) = i) y(\tau_k)) + (\tau_k - \tau_{k-1}) \cdot (x'_{i,j}(\tau_{k-1}^+) + \mathbf{1}(a(\tau_{k-1}^+) = i) y'_j(\tau_{k-1}^+)) - \tau'_{k-1,j} \cdot (x_i(\tau_{k-1}, \theta) + \mathbf{1}(a(\tau_{k-1}) = i) y(\tau_{k-1}))]$$

$$(30)$$

Note that evaluation of the IPA gradient estimator in (30) requires event time and state derivatives provided by (26)-(29), which only need readily observable data (e.g., queue lengths, event times) and some rate information at events (e.g., job arrival rates $a(\tau_k)$, service rates $\beta(\tau_k)$) that can be measured. Section IV provides implementation details of this estimation algorithm taking advantage of the fact that all this information is directly observable on any underlying DES sample path.

B. IPA Estimator Unbiasedness

The unbiasedness of an IPA derivative $\partial Q_i(\theta)/\partial \theta_j$ has been shown to be ensured by the following two conditions (see [46], Lemma A2, p.70): (i) For every $\theta \in \Theta$ (where Θ is a closed bounded set), the sample derivative $\partial Q_i(\theta)/\partial \theta_j$ exists w.p.1, and (ii) W.p.1, the random function $Q_i(\theta)$ is Lipschitz continuous throughout Θ , and the (generally random) Lipschitz constant has a finite first moment. Regarding condition (i), the existence of the sample derivatives $\partial Q_i(\theta)/\partial \theta_j$ is guaranteed by Assumption 1 and the following additional assumption:

Assumption 2. (a) For every $\theta \in \Theta$, 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 $\{r_i(t)\}$, $\{\beta_i(t)\}$, have identical values during any open subinterval of $[0, T]$.

We point out that even if the conditions of Assumption 2 do not hold, it is possible to use one-sided derivatives and still carry out IPA, as in [30]. Consequently, establishing the unbiasedness of $\partial Q_i(\theta)/\partial \theta_j$ reduces to verifying the Lipschitz continuity of the sample function $Q_i(\theta)$ with appropriate Lipschitz constants. To establish this, we will need the following additional mild technical assumptions on processing rates and the number of exogenous events.

Assumption 3: The processes $\{\beta_i(t)\}$ are upper bounded w.p.1, i.e., there exists $B < \infty$ such that $\beta_i(t) \leq B$ w.p.1 for all $t \in [0, T]$ and $i = 1, 2, \dots, N$.

Assumption 4: Let N_1 be the number of exogenous events over $[0, T]$. Then $E[N_1] < \infty$.

The proof of unbiasedness relies on two Lemmas. First, Lemma 1 provides a bound for the expected total number of events in the interval $[0, T]$.

Lemma 1: Let N_T be the number of events occurring during interval $[0, T]$. Then $E[N_T] < \infty$.

Proof: See Appendix.

Using Lemma 1, we can further bound all state and event time derivatives, as shown in Lemma 2 below.

Lemma 2: For all $t \in [0, T]$, state perturbations $|x'_{i,j}(\theta, t)|$ and $|y'_j(\theta, t)|$ are bounded w.p. 1, and for all $k = 1, 2, \dots, N_T$, and $|\tau'_k|$ 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 the IPA estimators.

Theorem 1. Under Assumptions 1-4, the IPA estimator $\partial Q_i(\theta)/\partial \theta_j$ is an unbiased estimate of $dE[Q_i(\theta)]/d\theta_j$.

Proof: See Appendix.

IV. DES OPTIMIZATION

In this section, we describe how the IPA estimator we have derived for the SFM of Section II can be used to determine optimal lot sizes for the actual underlying DES. Since the SFM can only be simulated, we can apply our analysis in such a simulation setting and take advantage of the fact that a flow model is much simpler to simulate than its DES counterpart. Instead, however, we choose to apply the IPA estimator using actual data from an observed sample path of the DES, which leads to an approximate solution of the original problem. Note that the availability of such data

does not require any assumptions on the random processes involved, which would otherwise be necessary to generate the realizations necessary in implementing a simulation model. We also note that an analytical quantification of the relationship between the optimal performance obtained using the SFM and that corresponding to the original DES is possible for some simple systems as described in [47], but it is generally a challenging task. However, empirical evidence of the accuracy of such an approximation is provided in what follows.

Let $J_T^{DES}(\theta)$ be the expected total workload over $[0, T]$ of the actual DES whose SFM we developed in Section II, under a lot size parameter vector θ . We can use the following stochastic approximation algorithm to seek the value of θ that minimizes $J_T^{DES}(\theta)$:

$$\theta_{n+1} = \theta_n - \eta_n H_n(\theta_n, \omega_n^{DES}) \quad (31)$$

where $H_n(\theta_n, \omega_n^{DES})$ is an estimate of $dJ_T^{DES}(\theta)/d\theta$ at θ_n which is unavailable. Since we cannot obtain such an estimate from the DES itself, we use instead an IPA estimator $dJ_T^{SFM}(\theta)/d\theta$ from the SFM where $J_T^{SFM}(\theta) = E[\mathcal{L}(\theta; x(0), T)]$ with $\mathcal{L}(\theta; x(0), T)$ defined in (11), thus approximating the optimal solution. The IPA estimator $dJ_T^{SFM}(\theta)/d\theta$ is evaluated using the data observed over a specific estimation interval using the IPA estimation algorithm given in the previous section. It is then used in (31) to update θ , and the process repeats with the updated θ in the next estimation interval. To use the event-driven IPA estimation algorithm, we need to identify all events defined in Section II-A for the SFM with observable events in the real DES. For exogenous events, we monitor changes in arrival rates $r_i(t)$ and $\beta_i(t)$ in the DES through simple rate estimators periodically updated over t ; if $|r_i(t) - r_i(t - \Delta)| > \epsilon$ for some adjustable Δ and ϵ , we detect such changes and similarly for β_i . For switching events, we just have to count the number of jobs processed: once the server has served $\lceil \theta_i \rceil$ class i jobs continuously, a switching event will occur to the next class. For induced events and lot-forming events, we identify them by monitoring the time elapsed after switching to the current class and the number of jobs in the current queue: when either the changeover is finished or the current queue has accumulated enough jobs to form a lot, we have an induced event or lot-forming event respectively.

Figure 5 shows an example of applying our IPA estimate and (31) in optimizing a two-class lot-sizing problem (*not* its SFM counterpart). The jobs of both classes arrive according to Poisson processes with time-varying rates; in particular, the rates are piecewise constant with values uniformly distributed over $[\frac{1}{2.1}, \frac{1}{1.7}]$ and $[\frac{1}{1.9}, \frac{1}{1.3}]$, and the times between changes of the rates are both exponentially distributed with average 1500 sec. The processing times for both classes are piecewise constant functions, whose values are uniform random variables over $[0.4, 0.6]$ and $[0.7, 0.9]$, and the times between changes of the processing times are both exponentially distributed with average 800 sec. The estimation interval is 150 sec, and the server is scheduled in round-robin fashion between the two classes and the changeover times are 15 sec and 25 sec respectively. The actual objective function shown is obtained by exhaustively simulating the real DES over all (θ_1, θ_2) pairs,

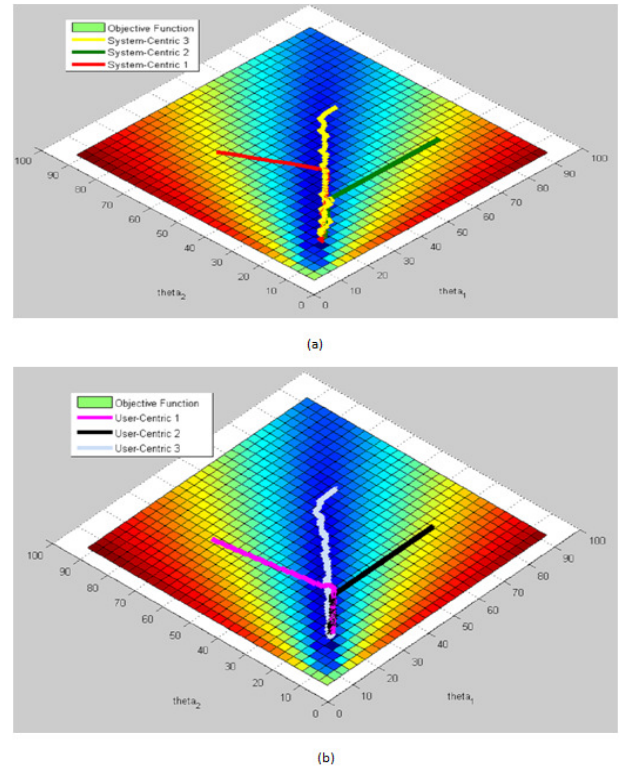


Fig. 5. (a) System-Centric Optimization (b) User-Centric Optimization

averaging over 50 sample paths for a time horizon of 4 hours. This gives (approximately) $\theta^* = (22, 16)$.

As mentioned earlier, an interesting aspect of this problem is that one can expect differences between a user-centric and system-centric optimization approach. In system-centric optimization, we use (10) as the objective function and Figure 5.(a) shows results in this case by implementing (31) using gradient estimates obtained by applying the IPA algorithm on a single sample path with different starting points. We can see that all results converge to a point very close to the “true” optimum, illustrating the effectiveness of our method. In the user-centric optimization, classes 1 and 2 individually optimize their own performance metric $J_i(\theta)$ by taking turns adjusting θ_i in a non-cooperative game setting. In this game, each class has no information on the other’s performance or control, i.e., class 1 updates θ_1 using $dJ_{1,T}^{SFM}(\theta)/d\theta_1$ evaluated during an estimation interval, while class 2 maintains its control θ_2 . Then, in the next estimation interval, class 1 uses its updated θ_1 , and class 2 takes its turn to evaluate $dJ_{2,T}^{SFM}(\theta)/d\theta_2$ and uses it to update θ_2 . Figure 5.(b) shows the results of user-centric optimization; comparing it to Figure 5.(a), we can see that (unlike previous resource contention problems in multiclass SFMs, e.g., [43], [44]) in the lot-sizing problem, there is no gap between system-centric and user-centric optimal costs.

Analyzing the difference between the two problem solutions under arbitrary arrival, setup, and service processes is extremely complicated. We have, however, in what follows, studied the deterministic version of the SFM in Figure 2 and can formally prove that system-centric and user-centric

optimization processes converge to the same point as shown next. In addition, the analysis therein can give us some insights on the reasons behind the “no-gap” property in the stochastic setting of this problem.

Theorem 2. Let θ_s^* and θ_u^* be the optimal lot sizes in the system-centric and user-centric optimization respectively. If $\{r_i\}, \{\beta_i\}$ and $\{s_i\}, i = 1, 2$, are all constant, then $\theta_s^* = \theta_u^*$.

Proof: See Appendix.

The intuition behind this “no-gap” property becomes clear when considering the three cases of system convergence analyzed in the theorem proof. Looking at the control space defined by (θ_1, θ_2) , there is only a subset over which the system can be stable, i.e., $x_1(t), x_2(t)$ remain finite. From the user-centric point of view, class 1 is unstable if θ_1 is too small because (i) there are too many changeovers and (ii) the system spends a relatively large time serving class 2; hence, its effective processing capacity is limited. However, if θ_1 is too large, then the length of both the input and the output queue of class 1 increases. Therefore, class 1 seeks the minimum possible value of θ_1 , for any given value of θ_2 , that allows it to be stable in order to optimize its own performance. A similar argument applies to class 2, whose optimal lot size is the smallest θ_2 that makes class 2 stable for a given θ_1 . Thus, user-centric optimization will lead us to the point where both classes are stable, and with smallest possible θ_1 and θ_2 . From the system’s point of view, in order to achieve the best possible overall performance, system-centric optimization will obviously also lead us to the region where both classes are stable. In addition, in that region, the performances of individual classes are decoupled (as explicitly shown in the proof) and the overall system performance is the sum of all class workloads, hence the result of system-centric optimization is also the point with smallest θ_1 and θ_2 in that region. Therefore, user-centric and system-centric optimization converge to the same lot size vector. It is also worth noticing that in the simulation results above, the optimal solution θ^* is indeed very close to the values given in (54) and (55) when r_i and β_i are replaced by their respective expected values. The same applies to the example discussed next.

Figure 6 shows an example of applying IPA and a similar optimization algorithm on the variant of the problem for a non-idling policy as discussed in Section II-B with the stochastic hybrid automaton model shown in Figure 4. One can see that the IPA algorithm works effectively for this variant as well, and similarly, there is no gap between system-centric and user-centric optimization.

V. CONCLUSIONS AND FUTURE WORK

We have developed a multiclass SFM for the lot-sizing problem based on which we derive IPA gradient estimators to drive an on-line optimization scheme for getting optimal lot sizes. In addition, similar to recent work on multiclass SFMs, we carry out optimization in both system-centric and user-centric manner. Interestingly, in the lot-sizing problem, these two forms of optimization lead to the same optimum of the performance function, as shown in our simulation results and by an analysis of a deterministic model. Our ongoing research

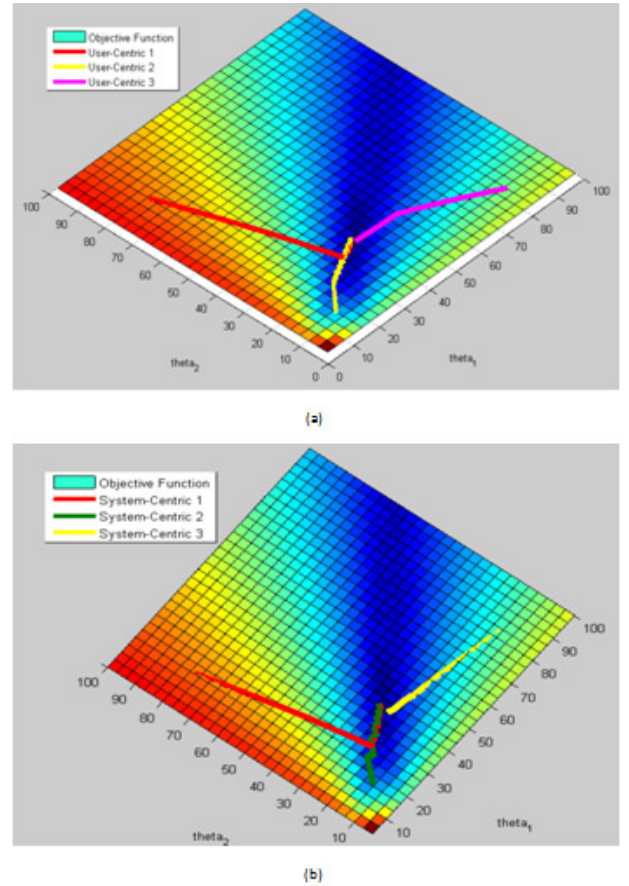


Fig. 6. Optimization Results for the Variant Model with Non-Idling Policy: (a) User-Centric Optimization (b) System-Centric Optimization

focuses on exploring the possibilities of extending our methods in the lot-sizing problem to more general problems involving resource switching control, as well as to multi-stage lot-sizing problems.

VI. APPENDIX

Proof of Lemma 1: Let N_1, N_2 and N_3 denote the number of exogenous events, switching events, and service start events respectively. First, based on Assumption 4,

$$E[N_1] \leq n_1 \quad (32)$$

for some positive number n_1 .

In order to study the number of switching events and service start events, we define a “cycle” to be the time interval between one service start event and the next service start event of the same class. Because of the round-robin scheduling policy, the server has to make N switches before it can start to serve the current class again, where N is the number of different classes. Therefore, for any sample path, during each cycle there are exactly N switching events and N service start events, hence

$$N_2 \leq N \cdot N_{c,T}, \quad N_3 \leq N \cdot N_{c,T} \quad (33)$$

where $N_{c,T}$ is the number of cycles over an interval $[0, T]$, which may also include an incomplete cycle at the end. Note that the length of a cycle, denoted by L_C , is comprised of

times spent in changeovers, together with actual job processing times and possibly times spent waiting for a sufficient number of jobs to form a lot. Therefore,

$$L_C > \sum_i^N s_i$$

and

$$N_{c,T} = \left\lfloor \frac{T}{L_C} \right\rfloor + 1 \leq \frac{T}{L_C} + 1 < \frac{T}{\sum_i^N s_i} + 1$$

Combining with (33), we get

$$N_2 < N \cdot \left(\frac{T}{\sum_i^N s_i} + 1 \right), \quad N_3 < N \cdot \left(\frac{T}{\sum_i^N s_i} + 1 \right)$$

Since $\{s_i\}$ are positive by definition, it follows that $E[N_2]$ and $E[N_3]$ are also bounded, i.e.,

$$\begin{aligned} E[N_2] &< n_2 \equiv N + NT \cdot E \left[\frac{1}{\sum_i^N s_i} \right] \\ E[N_3] &< n_3 \equiv N + NT \cdot E \left[\frac{1}{\sum_i^N s_i} \right] \end{aligned} \quad (34)$$

Since $N_T = N_1 + N_2 + N_3$, combining (32) and (34), we conclude that $E[N_T] < \infty$ and the result is established. ■

Proof of Lemma 2: First, using Lemma 1, we know that $E[N_T]$ is bounded, i.e., N_T is finite w.p.1. Since, from (18) and (19), we know that $x'_{i,j}(t)$ and $y'_j(t)$ are fixed between any two consecutive events, we only have to prove the boundness of $\{x'_{i,j}(\tau_k^+)\}$, $\{y'_j(\tau_k^+)\}$ and $\{\tau'_{k,j}\}$, for all $k \in \{0, 1, 2, \dots, N_T\}$. When $k = 0$, $x'_{i,j}(\tau_0^+) = 0$, $i = 1, 2$, and $y'_j(\tau_0^+) = \tau'_{0,j} = 0$, which are obviously bounded. Assume that for some integer n , $0 \leq n < N$, for all integers $k \leq n$, $|x'_{i,j}(\tau_k^+)|$, $|y'_j(\tau_k^+)|$ and $|\tau'_{k,j}|$ 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 (26), we know that

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

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

Case 2: A switching event occurs at τ_{n+1} . Then from (27), we have

$$\begin{aligned} x'_{i,j}(\tau_{n+1}^+) &= \\ \begin{cases} x'_{i,j}(\tau_{n+1}^-) + y'_j(\tau_{n+1}^-) - 1 & i = a(\tau_{n+1}^-), j = a(\tau_{n+1}^-) \\ x'_{i,j}(\tau_{n+1}^-) + y'_j(\tau_{n+1}^-) & i = a(\tau_{n+1}^-), j \neq a(\tau_{n+1}^-) \\ x'_{i,j}(\tau_{n+1}^-) & \text{otherwise} \end{cases} \\ y'_j(\tau_{n+1}^+) &= 0 \end{aligned}$$

Using the induction hypothesis, $|x'_{i,j}(\tau_{n+1}^+)|$ and $|y'_j(\tau_{n+1}^+)|$ are obviously bounded. Regarding $\tau'_{n+1,j}$, from (23),

$$\tau'_{n+1,j} = \begin{cases} (1 - y'_j(\tau_{n+1}^-)) / \beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-), & a(\tau_{n+1}^-) = j \\ -y'_j(\tau_{n+1}^-) / \beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-), & a(\tau_{n+1}^-) \neq j \end{cases}$$

and we have

$$|\tau'_{n+1,j}| \leq \frac{1 + |y'_j(\tau_{n+1}^-)|}{\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-)}$$

where $1 + |y'_j(\tau_{n+1}^-)| = 1 + |y'_j(\tau_n^+)|$ is bounded by the induction hypothesis, and $\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-)$ must be strictly positive, otherwise we cannot have $y(\tau_{n+1}^-) < \theta_{a(\tau_{n+1}^-)}$ and $y(\tau_{n+1}^+) = \theta_{a(\tau_{n+1}^+)}$ as required by a switching event at τ_{n+1} . Therefore, there exists a strictly positive number k_1 , such that $\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) > k_1$, and we get

$$|\tau'_{n+1,j}| \leq \frac{1 + |y'_j(\tau_{n+1}^-)|}{\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-)} \leq \frac{1 + |y'_j(\tau_{n+1}^-)|}{k_1}$$

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

Case 3: An induced event occurs at τ_{n+1} . First, from (25), the value of $\tau'_{n+1,j}$ is unchanged from the previous switching event, which is bounded by the induction hypothesis. Next, from (29),

$$\begin{aligned} x'_{i,j}(\tau_{n+1}^+) &= x'_{i,j}(\tau_{n+1}^-) + \beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) \cdot \tau'_{n+1,j} \\ &\quad \text{if } i = a(\tau_{n+1}^-) \\ x'_{i,j}(\tau_{n+1}^+) &= x'_{i,j}(\tau_{n+1}^-), \\ &\quad \text{if } i \neq a(\tau_{n+1}^-) \\ y'_j(\tau_{n+1}^+) &= y'_j(\tau_{n+1}^-) - \beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) \cdot \tau'_{n+1,j} \end{aligned}$$

Using the induction hypothesis together with Assumption 3, $|x'_{i,j}(\tau_{n+1}^+)|$ and $|y'_j(\tau_{n+1}^+)|$ are obviously also bounded.

Case 4: A lot-forming event occurs at τ_{n+1} . From (28), we have

$$\begin{aligned} x'_{i,j}(\tau_{n+1}^+) &= x'_{i,j}(\tau_{n+1}^-) + \frac{\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) \cdot (1 - x'_{i,j}(\tau_{n+1}^-))}{r_i(\tau_{n+1}^-)} \\ &\quad \text{if } i = a(\tau_{n+1}^-) = j \\ x'_{i,j}(\tau_{n+1}^+) &= x'_{i,j}(\tau_{n+1}^-) - \frac{\beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) \cdot x'_{i,j}(\tau_{n+1}^-)}{r_i(\tau_{n+1}^-)} \\ &\quad \text{if } i = a(\tau_{n+1}^-) \neq j \\ x'_{i,j}(\tau_{n+1}^+) &= x'_{i,j}(\tau_{n+1}^-) \\ &\quad \text{if } i \neq a(\tau_{n+1}^-) \\ y'_j(\tau_{n+1}^+) &= y'_j(\tau_{n+1}^-) - \beta_{a(\tau_{n+1}^-)}(\tau_{n+1}^-) \cdot \tau'_{n+1,j} \end{aligned}$$

It follows from these equations, together with Assumption 3, that

$$\begin{aligned} |x'_{i,j}(\tau_{n+1}^+)| &\leq |x'_{i,j}(\tau_n^+)| + B \cdot (1 + |x'_{i,j}(\tau_n^+)|) / r_i(\tau_{n+1}^-) \\ |y'_j(\tau_{n+1}^+)| &\leq |y'_j(\tau_n^+)| + B \cdot (1 + |x'_{i,j}(\tau_n^+)|) / r_i(\tau_{n+1}^-) \end{aligned}$$

where $|x'_{i,j}(\tau_n^+)|$ and $|y'_j(\tau_n^+)|$ are bounded by the induction hypothesis, and $r_i(\tau_{n+1}^-) > 0$, otherwise we cannot have

$x_i(\tau_{n+1}^-) < \theta_i$ and $x_i(\tau_{n+1}^+) = \theta_i$ as required by a lot-forming event at τ_{n+1} . Therefore, there exists a strictly positive number k_2 , such that $r_i(\tau_{n+1}) > k_2$, and we get

$$\begin{aligned} |x'_{i,j}(\tau_{n+1}^+)| &\leq |x'_{i,j}(\tau_n^+)| + B \cdot (1 + |x'_{i,j}(\tau_n^+)|) / k_2 \\ |y'_{i,j}(\tau_{n+1}^+)| &\leq |y'_{i,j}(\tau_n^+)| + B \cdot (1 + |x'_{i,j}(\tau_n^+)|) / k_2 \end{aligned}$$

which implies that $|x'_{i,j}(\tau_{n+1}^+)|$ and $|y'_{i,j}(\tau_{n+1}^+)|$ are also bounded.

This completes the inductive step and, recalling that N_T is finite w.p.1, the result is established. ■

Proof of Theorem 1: Invoking Lemma A2 from [46], the unbiasedness of the IPA estimator $\partial Q_i(\theta)/\partial \theta_j$ relies on the Lipschitz continuity of $Q_i(\theta)$. From (12),

$$Q_i(\theta) = \frac{1}{T} \int_0^T (x_i(t, \theta) + \mathbf{1}(a(t) = i) \cdot y(t)) dt$$

and taking derivatives with respect to θ_j yields

$$\begin{aligned} |\partial Q_i(\theta)/\partial \theta_j| &= \frac{1}{T} \left| \int_0^T (x'_{i,j}(t) + \mathbf{1}(a(t) = i) \cdot y'_j(t)) dt \right| \\ &\leq \frac{1}{T} \int_0^T (|x'_{i,j}(t)| + |y'_j(t)|) dt \end{aligned}$$

Based on Lemma 2, we know that $|x'_{i,j}(t)|$ and $|y'_j(t)|$ are bounded, therefore, the above inequality guarantees the boundedness of $|\partial Q_i(\theta)/\partial \theta_j|$, i.e.,

$$|\partial Q_i(\theta)/\partial \theta_j| < B_Q < \infty$$

where B_Q is such that $\frac{1}{T} \int_0^T (|x'_{i,j}(t)| + |y'_j(t)|) dt < B_Q$, and we get

$$|\Delta Q_i(\theta)| = |\partial Q_i(\theta)/\partial \theta_j| |\Delta \theta_j| < B_Q |\Delta \theta_j| \quad (35)$$

Thus, $Q_i(\theta)$ is Lipschitz continuous with finite Lipschitz constant B_Q . ■

Proof of Theorem 2: Let us partition the state trajectory of the system into ‘‘cycles’’ as defined in the proof of Lemma 1 (i.e., time intervals between a service start event and the next service start event of the same class). Let $l_{1,k}(\theta_1)$ and $l_{2,k}(\theta_2)$ denote the length of time spent serving class 1 and class 2 respectively within the k th cycle, $k = 1, 2, \dots$, which obviously depends on the lot sizes θ_1, θ_2 . In addition, let $x_{i,k}^s$ be the class i content at the time when the server is switched to class $i = 1, 2$ within the k th cycle. Then, the net class 1 and class 2 flow volumes processed in the k th cycle are:

$$\begin{aligned} \Delta x_{1,k}(\theta_1, \theta_2) &= x_{1,k+1}^s - x_{1,k}^s \\ &= r_1 \cdot [l_{1,k}(\theta_1) + l_{2,k}(\theta_2)] - \theta_1 \\ \Delta x_{2,k}(\theta_1, \theta_2) &= x_{2,k+1}^s - x_{2,k}^s \\ &= r_2 \cdot [l_{1,k+1}(\theta_1) + l_{2,k}(\theta_2)] - \theta_2 \end{aligned} \quad (36)$$

Clearly, if θ_i is such that $\Delta x_{i,k}(\theta_1, \theta_2) > 0$, then $x_{i,k}^s$ is increasing and the class i queue is obviously unstable and not optimal. If on the other hand $\Delta x_{i,k}(\theta_1, \theta_2) \leq 0$, then $x_{i,k+1}^s \leq x_{i,k}^s$. To analyze the exact behavior of $x_{i,k}^s$ over $k = 1, 2, \dots$ let us start with $k = 1$ and assume the system starts out empty

with the server switched to class 1. If $x_{1,1}^s + r_1 s_1 = r_1 s_1 < \theta_1$, then class 1 cannot form a lot by the time the changeover is done at $t = s_1$, therefore, there is an additional time interval given by $w_{1,1} = \frac{\theta_1}{r_1} - s_1$ until a lot-forming event occurs for class 1. It follows that

$$l_{1,1} = s_1 + w_{1,1} + \frac{\theta_1}{\beta_1} = \frac{\theta_1}{r_1} + \frac{\theta_1}{\beta_1}$$

When the server switches to class 2, we have $x_{2,1}^s = r_2 \cdot l_{1,1}$. If $x_{2,1}^s + r_2 s_2 < \theta_2$, then class 2 must also wait for a lot-forming event and

$$l_{2,1} = \frac{\theta_2}{r_2} - l_{1,1} + \frac{\theta_2}{\beta_2} \quad (37)$$

In the second cycle, $x_{1,2}^s = r_1 \cdot \left(\frac{\theta_1}{\beta_1} + l_{2,1} \right)$. If $x_{1,2}^s + r_1 s_1 = r_1 \cdot \left(\frac{\theta_1}{\beta_1} + l_{2,1} + s_1 \right) < \theta_1$, then there is once again an interval $w_{1,2} = \frac{\theta_1}{r_1} - \left(l_{2,1} + \frac{\theta_1}{\beta_1} + s_1 \right)$ until a class 1 lot-forming event and we get

$$l_{1,2} = s_1 + w_{1,2} + \frac{\theta_1}{\beta_1} = \frac{\theta_1}{r_1} - l_{2,1}$$

For class 2, using (36) and (37), we have $x_{2,2}^s = r_2 \cdot \left(\frac{\theta_2}{\beta_2} + l_{1,2} \right)$. Similarly, if $x_{2,2}^s + r_2 s_2 = r_2 \cdot \left(\frac{\theta_2}{\beta_2} + l_{1,2} + s_2 \right) < \theta_2$, there is a time interval $w_{2,2} = \frac{\theta_2}{r_2} - \left(\frac{\theta_2}{\beta_2} + l_{1,2} + s_2 \right)$ before a lot-forming event occurs for class 2, and

$$l_{2,2} = s_2 + w_{2,2} + \frac{\theta_2}{\beta_2} = \frac{\theta_2}{r_2} - l_{1,2} = \left(\frac{\theta_2}{r_2} - \frac{\theta_1}{r_1} \right) + l_{2,1}$$

Thus, $l_{2,2} - l_{2,1} = \frac{\theta_2}{r_2} - \frac{\theta_1}{r_1}$. Following the same series of steps over $k = 1, 2, \dots$, as long as $x_{i,m}^s + r_i s_i < \theta_i$ for all $m \leq k$, $i = 1, 2$, we obtain:

$$x_{1,k}^s = r_1 \cdot \left(\frac{\theta_1}{\beta_1} + l_{2,k-1} \right), \quad x_{2,k}^s = r_2 \cdot \left(\frac{\theta_2}{\beta_2} + l_{1,k} \right) \quad (38)$$

and

$$\begin{aligned} l_{1,k} &= \frac{\theta_1}{r_1} - l_{2,k-1} = \left(\frac{\theta_1}{r_1} - \frac{\theta_2}{r_2} \right) + l_{1,k-1} \\ l_{2,k} &= \frac{\theta_2}{r_2} - l_{1,k} = \left(\frac{\theta_2}{r_2} - \frac{\theta_1}{r_1} \right) + l_{2,k-1} \end{aligned} \quad (39)$$

Next, we proceed by considering the three possible cases defined by the sign of $\frac{\theta_1}{r_1} - \frac{\theta_2}{r_2}$:

Case 1: If $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$, then based on (39), $l_{1,k} > l_{1,k-1}$ and $l_{2,k} < l_{2,k-1}$, i.e., $\{l_{1,k}\}$ increases and $\{l_{2,k}\}$ decreases, hence, by (38), $\{x_{1,k}^s\}$ decreases and $\{x_{2,k}^s\}$ increases. Therefore, there must exist some $n \in \mathbb{Z}^+$, such that $x_{2,n}^s + r_2 s_2 \geq \theta_2$ and $x_{2,n-1}^s + r_2 s_2 < \theta_2$, which implies that the n th cycle is the first one such that class 2 can form a lot as soon as the changeover is done. Thus,

$$l_{2,n} = s_2 + \frac{\theta_2}{\beta_2} < s_2 + \frac{\theta_2}{\beta_2} + w_{2,n-1} = l_{2,n-1} \quad (40)$$

In the $n+1$ th cycle, by (38), $x_{1,n+1}^s = r_1 \cdot \left(\frac{\theta_1}{\beta_1} + l_{2,n} \right)$, and, based on (40), $x_{1,n+1}^s < r_1 \cdot \left(\frac{\theta_1}{\beta_1} + l_{2,n-1} \right) = x_{1,n}^s$. Therefore, $x_{1,n+1}^s + r_1 s_1 < x_{1,n}^s + r_1 s_1 < \theta_1$, so that class 1 will still

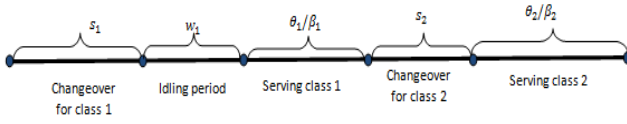


Fig. 7. Converging Cycle when $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$

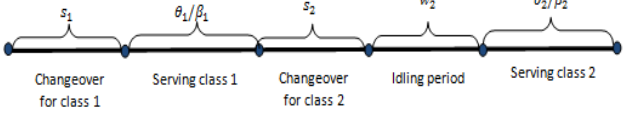


Fig. 8. Converging Cycle when $\frac{\theta_1}{r_1} < \frac{\theta_2}{r_2}$

includes a lot-forming event in this cycle, which from (39) implies that $l_{1,n+1} = \frac{\theta_1}{r_1} - l_{2,n}$. Using (40) again, $l_{1,n+1} > \frac{\theta_1}{r_1} - l_{2,n-1} = l_{1,n}$, hence $x_{2,n+1}^s + r_2 s_2 = r_2 \cdot \left(\frac{\theta_2}{\beta_2} + l_{1,n+1} \right) + r_2 s_2 > r_2 \cdot \left(\frac{\theta_2}{\beta_2} + l_{1,n} \right) + r_2 s_2 = x_{2,n}^s + r_2 s_2 \geq \theta_2$. Therefore, in the $n+1$ th cycle, class 2 can also form a lot as soon as the changeover is done and $l_{2,n+1} = l_{2,n} = s_2 + \frac{\theta_2}{\beta_2}$. By repeating the above analysis, it is easy to see that for all subsequent cycles $k \geq n+1$, $\{x_{1,k}^s\}$, $\{l_{1,k}\}$ and $\{l_{2,k}\}$ will remain fixed, i.e.,

$$\begin{aligned} x_{1,k}^s &= r_1 \cdot \left(s_2 + \frac{\theta_2}{\beta_2} + \frac{\theta_1}{\beta_1} \right) \\ l_{1,k} &= \frac{\theta_1}{r_1} - s_2 - \frac{\theta_2}{\beta_2}, \quad l_{2,k} = s_2 + \frac{\theta_2}{\beta_2} \end{aligned} \quad (41)$$

and since $l_{1,k} + l_{2,k} = \frac{\theta_1}{r_1}$, then using (36) together with the assumption $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$, we have $\Delta x_{2,k}(\theta_1, \theta_2) > 0$ for all $k \geq n+1$, which implies that the class 2 queue is unstable. The structure of a typical cycle $k \geq n+1$ in this case is shown in Figure 7.

Case 2: If $\frac{\theta_1}{r_1} < \frac{\theta_2}{r_2}$, following a similar analysis as in *Case 1*, there exists some $n' \in \mathbb{Z}^+$ such that for all $k > n'$:

$$\begin{aligned} x_{2,k}^s &= r_2 \cdot \left(s_1 + \frac{\theta_2}{\beta_2} + \frac{\theta_1}{\beta_1} \right) \\ l_{1,k} &= s_1 + \frac{\theta_1}{\beta_1}, \quad l_{2,k} = \frac{\theta_2}{r_2} - s_1 - \frac{\theta_1}{\beta_1} \end{aligned} \quad (42)$$

and the class 1 queue is unstable. The structure of a typical cycle $k \geq n+1$ in this case is shown in Figure 8.

Case 3: If $\frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, based on (39), $l_{1,k}$ and $l_{2,k}$ will be constant for all k . There are three possible subcases based on

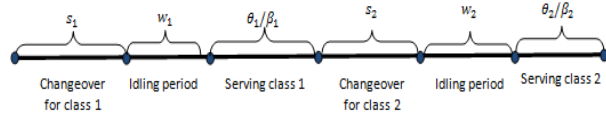


Fig. 9. Converging Cycle when $\frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$ and $\sum_{i=1}^2 s_i + \frac{\theta_i}{\beta_i} \leq \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$

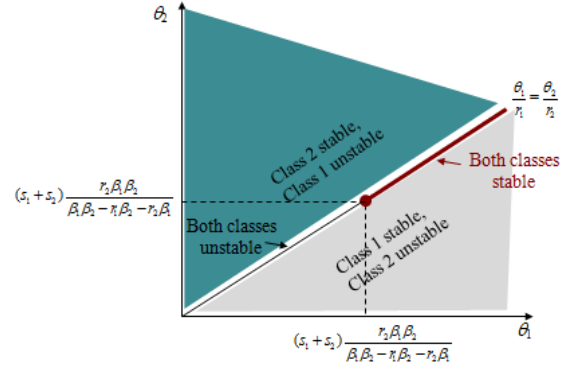


Fig. 10. Partition of the (θ_1, θ_2) feasible space

the (fixed) value of $l_{1,k} + l_{2,k}$:

(i) If $l_{1,k} + l_{2,k} > \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, then according to (36), both classes will become unstable, and $l_{i,k} = s_i + \frac{\theta_i}{\beta_i}$, $i = 1, 2$.

(ii) If $l_{1,k} + l_{2,k} < \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, then using (38), $x_{1,k}^s + r_1 s_1 = r_1 \left(s_1 + \frac{\theta_1}{\beta_1} + l_{2,k-1} \right) = r_1 \left(s_1 + \frac{\theta_1}{\beta_1} + l_{2,k} \right) \leq r_1 (l_{1,k} + l_{2,k}) < r_1 \cdot \frac{\theta_1}{r_1} = \theta_1$, which implies that class 1 has a lot-forming event in this cycle, and we can use (39) to get $l_{1,k} = \frac{\theta_1}{r_1} - l_{2,k-1} = \frac{\theta_1}{r_1} - l_{2,k}$, which contradicts our assumption that $l_{1,k} + l_{2,k} < \frac{\theta_1}{r_1}$. Therefore, this subcase is infeasible.

(iii) If $l_{1,k} + l_{2,k} = \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, then, since $l_{i,k} \geq s_i + \frac{\theta_i}{\beta_i}$, $i = 1, 2$, we must have $\sum_{i=1}^2 \left(s_i + \frac{\theta_i}{\beta_i} \right) \leq \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, which can be further reduced to, for $i = 1, 2$,

$$\theta_i \geq (s_1 + s_2) \left(\frac{r_i \beta_1 \beta_2}{\beta_1 \beta_2 - r_1 \beta_2 - r_2 \beta_1} \right) \quad (43)$$

The structure of a typical cycle in this case is shown in Figure 9, where $w_i = 0$ when $l_{i,k} = s_i + \frac{\theta_i}{\beta_i}$.

From the above three cases, we conclude that if $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$, the system converges to cycles as shown in Figure 7 where the class 1 queue satisfies (41) while the class 2 queue is unstable; similarly, if $\frac{\theta_1}{r_1} < \frac{\theta_2}{r_2}$, the system converges to cycles as shown in Fig. 8 where the class 2 queue satisfies (42) while the class 1 queue is unstable. Finally, if $\frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, then the system is either unstable for both classes when $\sum_{i=1}^2 s_i + \frac{\theta_i}{\beta_i} > \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, or it converges to cycles as shown in Figure 9 when θ_i satisfies (43). Figure 10 shows the partition of the two-dimensional feasible space of lot size values (θ_1, θ_2) based on this analysis.

Now let us study the effect of θ_i on the performance of class i concentrating on cycles k such that all convergent behavior

has been attained. When $i = 1$, for a fixed θ_2 , if $\frac{\theta_1}{r_1} < \frac{\theta_2}{r_2}$ or $\sum_{i=1}^2 \left(s_i + \frac{\theta_i}{\beta_i} \right) > \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, the class 1 queue becomes unstable. When $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$, or $\frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$ and (43) holds, the class 1 queue is stable. Let $F_1^+(\theta_2) \subseteq \mathbb{R}^+$ denote the region of feasible θ_1 values where the class 1 queue is stable for some

fixed θ_2 , i.e.,

$$F_1(\theta_2) = \begin{cases} \left(\frac{r_1\theta_2}{r_2}, \infty \right) & \text{if } \theta_2 < \frac{(s_1+s_2) \cdot r_2\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \\ \left[\frac{r_1\theta_2}{r_2}, \infty \right) & \text{if } \theta_2 \geq \frac{(s_1+s_2) \cdot r_2\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \end{cases} \quad (44)$$

Let $x_{1,lf}$ denote the class 1 queue content when a lot-forming event occurs in a cycle k after convergent behavior is attained. By the definition of a lot-forming event, we have $x_{1,lf} = \theta_1$. Let $x_{1,sw}$ denote the class 1 queue content when the next switching event occurs in the same cycle after the class 1 lot is processed, so that $x_{1,sw} = r_1 \frac{\theta_1}{\beta_1}$. The length of the interval between the lot-forming event and the switching event is $\frac{\theta_1}{\beta_1}$, while the length of the interval between this switching event and the lot-forming event in the next cycle is $l_{2,k} + s_1 + w_{1,k+1}$. Since the dynamics of x_1 in (1) over both of these intervals are linear, the time average of the class 1 workload, \bar{x}_1 is given by

$$\begin{aligned} \bar{x}_1 &= \frac{\frac{x_{1,lf} + x_{1,sw}}{2} \cdot \frac{\theta_1}{\beta_1} + \frac{x_{1,lf} + x_{1,sw}}{2} \cdot (s_1 + l_{2,k} + w_{1,k+1})}{\frac{\theta_1}{\beta_1} + l_{2,k} + s_1 + w_{1,k+1}} \\ &= \frac{1}{2} (x_{1,lf} + x_{1,sw}) = \frac{(\beta_1 + r_1)\theta_1}{2\beta_1} \end{aligned} \quad (45)$$

Recalling the dynamics of the class 1 output queue in (2), we have $\mathbf{1}(a(t) = 1) \cdot y(t) > 0$ in (12) only during the processing of class 1 in each cycle and the time average of this queue length is

$$\begin{aligned} \bar{y}_1 &= \frac{\frac{1}{2}(0 + \theta_1) \cdot \frac{\theta_1}{\beta_1}}{l_{1,k} + l_{2,k}} \\ &= \frac{\frac{\theta_1}{2} \cdot \frac{\theta_1}{\beta_1}}{\frac{\theta_1}{r_1}} = \frac{r_1\theta_1}{2\beta_1} \end{aligned} \quad (46)$$

Based on (45) and (46) and recalling (12), the performance function $J_1(\theta_1, \theta_2)$ in this case becomes:

$$\begin{aligned} J_1(\theta_1, \theta_2) &= E[Q_1(\theta_1, \theta_2)] \\ &= \frac{1}{T} \int_0^T (x_1(t, \theta) + \mathbf{1}(a(t) = 1) \cdot y(t)) dt \\ &= \frac{(\beta_1 + 2r_1)\theta_1}{2\beta_1} \end{aligned} \quad (47)$$

which is an increasing function of θ_1 . Therefore, for a fixed θ_2 , $J_1(\theta_1)$ is simply minimized by

$$\begin{aligned} \arg \min_{\theta_1} J_1(\theta_1, \theta_2) &= \arg \min_{\theta_1 \in F_1(\theta_2)} \frac{(\beta_1 + 2r_1)\theta_1}{2\beta_1} \\ &= \min\{\theta_1 : \theta_1 \in F_1(\theta_2)\} \end{aligned} \quad (48)$$

Along the same lines, for a fixed θ_1 , when $\frac{\theta_1}{r_1} > \frac{\theta_2}{r_2}$ or $\sum_{i=1}^2 \left(s_i + \frac{\theta_i}{\beta_i} \right) > \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}$, the class 2 queue becomes unstable, otherwise it is stable and we can define $F_2(\theta_1)$ as the region of θ_2 where the class 2 queue is stable for a given θ_1 , i.e.,

$$F_2(\theta_1) = \begin{cases} \left(r_2 \cdot \frac{\theta_1}{r_1}, \infty \right) & \text{if } \theta_1 < \frac{(s_1+s_2) \cdot r_1\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \\ \left[r_2 \cdot \frac{\theta_1}{r_1}, \infty \right) & \text{if } \theta_1 \geq \frac{(s_1+s_2) \cdot r_1\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \end{cases} \quad (49)$$

Then, using the same analysis for deriving (47), the performance function of class 2 in $F_2(\theta_1)$ is

$$J_2(\theta_1, \theta_2) = \frac{(\beta_2 + 2r_2)\theta_2}{2\beta_2} \quad (50)$$

the performance of class 2 is optimized by

$$\begin{aligned} \arg \min_{\theta_2} J_2(\theta_1, \theta_2) &= \arg \min_{\theta_2 \in F_2(\theta_1)} \frac{(\beta_2 + 2r_2)\theta_2}{2\beta_2} \\ &= \arg \min\{\theta_2 : \theta_2 \in F_2(\theta_1)\} \end{aligned} \quad (51)$$

Suppose the solution of the user-centric optimization problem is $\theta^* = (\theta_1^*, \theta_2^*)$. Then, both classes should be stable at this point, otherwise, the lot size of the unstable class cannot have an equilibrium value at this point. From the above analysis and referring to Figure 10, the region where both classes are stable, denoted by $F(\theta)$, is given by

$$F(\theta) = \quad (52)$$

$$\left\{ \theta \in \mathbb{R}^2 : \frac{\theta_1}{r_1} = \frac{\theta_2}{r_2}, \theta_i \geq \frac{(s_1 + s_2) \cdot r_i\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1}, i = 1, 2 \right\}$$

On the other hand, from the definition of the game, we have

$$\frac{\partial J_1}{\partial \theta_1}(\theta_1^*, \theta_2^*) = 0, \quad \frac{\partial J_2}{\partial \theta_2}(\theta_1^*, \theta_2^*) = 0 \quad (53)$$

which implies that when θ_2 is fixed to θ_2^* , then θ_1^* minimizes J_1 . Then, based on (48) and (52), we have

$$\begin{aligned} \theta_1^* &= \min\{\theta_1 : \theta_1 \in F_1(\theta_2^*), (\theta_1, \theta_2^*) \in F(\theta)\} \\ &= (s_1 + s_2) \cdot \frac{r_1\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \end{aligned} \quad (54)$$

and, similarly,

$$\begin{aligned} \theta_2^* &= \min\{\theta_2 : \theta_2 \in F_2(\theta_1^*), (\theta_1^*, \theta_2) \in F(\theta)\} \\ &= (s_1 + s_2) \cdot \frac{r_2\beta_1\beta_2}{\beta_1\beta_2 - r_1\beta_2 - r_2\beta_1} \end{aligned} \quad (55)$$

For a perturbation $\Delta\theta_2 > 0$ of θ_2 , first consider the point $(\theta_1^*, \theta_2^* - \Delta\theta_2)$. Since $\theta_1^* = \frac{r_1\theta_2^*}{r_2} > \frac{r_1(\theta_2^* - \Delta\theta_2)}{r_2}$ and $\theta_1^* \in F_1(\theta_2^* - \Delta\theta_2)$, by (45), $J_1(\theta_1)$ is the same as that of the point (θ_1^*, θ_2^*) , i.e.,

$$J_1(\theta_1^*, \theta_2^*) - J_1(\theta_1^*, \theta_2^* - \Delta\theta_2) = 0$$

therefore,

$$\lim_{\Delta\theta_2 \rightarrow 0} \frac{J_1(\theta_1^*, \theta_2^*) - J_1(\theta_1^*, \theta_2^* - \Delta\theta_2)}{\Delta\theta_2} = 0 \quad (56)$$

Next, for the point $(\theta_1^*, \theta_2^* + \Delta\theta_2)$, we have $\theta_1^* = \frac{r_1\theta_2^*}{r_2} < \frac{r_1(\theta_2^* + \Delta\theta_2)}{r_2}$, so that the class 1 queue falls into the unstable region, and

$$J_1(\theta_1^*, \theta_2^* + \Delta\theta_2) - J_1(\theta_1^*, \theta_2^*) > 0$$

thus

$$\lim_{\Delta\theta_2 \rightarrow 0} \frac{J_1(\theta_1^*, \theta_2^* + \Delta\theta_2) - J_1(\theta_1^*, \theta_2^*)}{\Delta\theta_2} > 0 \quad (57)$$

Combining (53), (56) and (57), it is clear that (θ_1^*, θ_2^*) is the minimum of $J_1(\theta_1, \theta_2)$. Similarly, we can show that (θ_1^*, θ_2^*) is also the minimum of $J_2(\theta_1, \theta_2)$. Therefore, (θ_1^*, θ_2^*) given by (54) and (55) is the optimal point of the overall performance function $J(\theta_1, \theta_2) = J_1(\theta_1, \theta_2) + J_2(\theta_1, \theta_2)$, i.e., the solution of the system-centric optimization problem. ■

REFERENCES

- [1] H. Takagi, "Analysis and application of polling models," *Lecture notes in computer science*, vol. 1769, pp. 423–442, 2000.
- [2] S. Fuhrmann, "Symmetric queues served in cyclic order," *Operation Research Letter*, vol. 4, no. 3, pp. 139–144, October 1985.
- [3] J. Hayes and J. Nadushan, "Numerical solution to limited service polling models," *Computer Communication*, vol. 9, no. 4, pp. 171–176, August 1986.
- [4] S. Fuhrmann and Y. Wang, *Mean Waiting time approximations of cyclic service systems with limited service*. Elsevier North-Holland, Amsterdam, 1988, pp. 253–265.
- [5] K. Leung, "Cyclic-service systems with nonpreemptive, time-limited service," *IEEE Transactions on Communications*, vol. 42, no. 8, pp. 2521–2524, August 1994.
- [6] F. Harris, *Operations and Cost*, ser. Factory Management Series. A.W. Shaw Co. Chicago, 1915.
- [7] G. Tinarelli, "Inventory control: Models and problems," *European Journal of Operational Research*, vol. 14, no. 1, pp. 1–12, 1983.
- [8] M. Salomon, *Deterministic lot sizing models for production planning*. Springer, Inc., 1991.
- [9] K. Baker, P. Dixon, M. Magazine, and E. Silver, "An algorithm for the dynamic lot-size problem with time-varying production capacity constraints," *Management Science*, vol. 24, no. 16, pp. 1710–1720, 1978.
- [10] L. Wolsey, *Integer Programming*. John Wiley and Sons, Inc., 1998.
- [11] M. Florian, J. Lenstra, and H. Rinnooy, "Deterministic production planning: Algorithms and complexity," *Management Science*, vol. 46, no. 5, pp. 669–679, 1980.
- [12] J. Maes and L. Wassenhove, "Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review," *Journal of Operation Research*, vol. 39, no. 11, pp. 991–1004, 1988.
- [13] N. Absi and S. Kedad-Sidhoum, "MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs," *Operations Research*, vol. 41, no. 2, pp. 171–192, 2007.
- [14] G. Belvaux and L. Wolsey, "bc-prod: A specialized branch-and-cut system for lot-sizing problems," *Management Science*, vol. 46, no. 5, pp. 724–738, 2000.
- [15] K. Akartunah and A. J. Miller, "A heuristic approach for big bucket multi-level production planning problems," *European Journal of Operational Research*, vol. 193, no. 2, pp. 396–411, 2009.
- [16] M. T. A. Federgruen, J. Meissner, "Progressive interval heuristics for multi-item capacitated lot-sizing problems," *Operations Research*, vol. 55, no. 3, pp. 490–502, 2007.
- [17] J. M. W. B. Karimi, S. M. T. F. Ghomiand, "A tabu search heuristic for solving the clsp with backlogging and set-up carry-over," *Journal of the Operational Research Society*, vol. 57, no. 2, pp. 140–147, 2006.
- [18] Y. L. E. P. S. Kim, J. Han, "Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum," *Computers and Operations Research*, vol. 37, no. 2, pp. 302–314, 2010.
- [19] Y. P. S. Kucukyavuz, "Uncapacitated lot sizing with backlogging: the convex hull," *Mathematical Programming*, vol. 118, no. 1, pp. 151–175, 2009.
- [20] P. Afentakis, B. Gavish, and U. Karmarkar, "Exact solutions to the lot-sizing problem in multi-stage assembly systems," *Management Science*, vol. 30, no. 1, pp. 222–239, 1984.
- [21] N. Simpson and S. Erengue, "Modeling multiple stage manufacturing systems with generalized costs and capacity issues," *Naval Research Logistics*, vol. 52, no. 6, pp. 560–570, 2005.
- [22] H. T. S. H. F. Sahling, L. Buschkuhl, "Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic," *Computers and Operations Research*, vol. 36, no. 9, pp. 2546–2553, 2009.
- [23] H. Sadtler, "Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows," *Operations Research*, vol. 51, no. 3, pp. 487–502, 2003.
- [24] L. B. H. Tempelmeier, "A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures," *OR Spectrum*, vol. 31, no. 2, pp. 385–404, 2009.
- [25] U. Karmarkar, S. Kekre, and S. Freeman, "Lot-sizing and lead-time performance in a manufacturing cell," *Interfaces*, vol. 15, no. 2, pp. 1–9, 1985.
- [26] S. Graves, A. Rinnooy, and P. Zipkin, *Logistics of Production and Inventory*. North-Holland, 1993.
- [27] H. Hafner, "Lot-sizing and throughput times in a job shop," *International Journal of Production Economics*, vol. 23, no. 3, pp. 111–116, 1991.
- [28] U. Karmarkar, "Lot sizes, lead times and in-process inventories," *Management Science*, vol. 33, no. 3, pp. 409–418, 1987.
- [29] C. G. Cassandras and R. Yu, "A 'surrogate problem' approach for lot size optimization in manufacturing systems," *Proc. of 2000 American Control Conference*, pp. 3279–3283, 2000.
- [30] C. G. Cassandras, Y. Wardi, B. Melamed, G. Sun, and C. G. Panayiotou, "Perturbation analysis for on-line control and optimization of stochastic fluid models," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1234–1248, 2002.
- [31] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *The Bell System Technical Journal*, vol. 61, pp. 1871–1894, 1982.
- [32] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. B. Gong, "Fluid simulation of large scale networks: Issues and tradeoffs," in *Proceedings of the Intl. Conf. on Parallel and Distributed Processing Techniques and Applications*, June 1999, pp. 2136–2142.
- [33] D. Connor, G. Feigin, and D. D. Yao, "Scheduling semiconductor lines using a fluid network model," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 88–98, 1994.
- [34] A. Elwalid and D. Mitra, "Analysis and design of rate-based congestion control of high-speed networks, i: Stochastic fluid models, access regulation," *Queueing Systems*, vol. 9, pp. 29–64, 1991.
- [35] N. G. S. R. Mahabhashyam and S. Kumara, "Resource-sharing queueing systems with fluid-flow traffic," *Operations Research*, vol. 56, no. 3, pp. 728–744, 2008.
- [36] A. Yan and W. B. Gong, "Fluid simulation for high-speed networks with flow-based routing," *IEEE Transactions on Information Theory*, vol. 45, pp. 1588–1599, 1999.
- [37] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation analysis and optimization of stochastic hybrid systems," *European Journal of Control*, vol. 16, no. 6, pp. 642–664, 2010.
- [38] C. G. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems, 2nd Edition*. Springer, 2008.
- [39] H. Yu and C. Cassandras, "Perturbation analysis of feedback-controlled stochastic flow systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1317–1332, 2004.
- [40] Y. Wardi, R. Adams, and B. Melamed, "A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 89–103, 2009.
- [41] G. Sun, C. G. Cassandras, and C. G. Panayiotou, "Perturbation analysis and optimization of stochastic flow networks," *IEEE Transactions Automatic Control*, vol. 49, no. 12, pp. 2113–2128, 2004.
- [42] H. Yu and C. Cassandras, "Perturbation analysis and feedback control of communication networks using stochastic hybrid models," *Journal of Nonlinear Analysis*, vol. 65, no. 6, pp. 1251–1280, 6 2006.
- [43] C. Yao and C. Cassandras, "Perturbation analysis and optimization of multiclass multiobjective stochastic flow models," *Journal of Discrete Event Dynamic Systems*, vol. 21, no. 2, pp. 219–256, 2011.
- [44] —, "Perturbation analysis and resource contention games in multiclass stochastic fluid models," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 301–319, 2011.
- [45] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.
- [46] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. New York, New York: John Wiley and Sons, 1993.
- [47] C. Yao and C. G. Cassandras, "Using infinitesimal perturbation analysis of stochastic flow models to recover performance sensitivity estimates of discrete event systems," in *Proceedings of 18th IFAC World Congress*, 2011, pp. 8217–8222.