MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 11, No. 1, Winter 2009, pp. 70–89
ISSN 1523-4614 | EISSN 1526-5498 | 09 | 1101 | 0070

DOI 10.1287/msom.1070.0195
© 2009 INFORMS

Pointwise Stationary Fluid Models for Stochastic Processing Networks

Achal Bassamboo
Kellogg School of Management, Northwestern University, Evanston, Illinois 60208,
a-bassamboo@nwu.edu

J. Michael Harrison
Graduate School of Business, Stanford University, Stanford, California 94305,
harrison_michael@gsb.stanford.edu

Assaf Zeevi
Graduate School of Business, Columbia University, New York, New York 10027,
assaf@gsb.columbia.edu

Generalizing earlier work on staffing and routing in telephone call centers, we consider a processing network model with large server pools and doubly stochastic input flows. In this model the processing of a job may involve several distinct operations. Alternative processing modes are also allowed. Given a finite planning horizon, attention is focused on the two-level problem of capacity choice and dynamic system control. A pointwise stationary fluid model (PSFM) is used to approximate system dynamics, which allows development of practical policies with a manageable computational burden. Earlier work in more restrictive settings suggests that our method is asymptotically optimal in a parameter regime of practical interest, but this paper contains no formal limit theory. Rather, it develops a PSFM calculus that is broadly accessible, with an emphasis on modeling and practical computation.

Key words: admission control; dynamic routing; doubly stochastic arrivals; approximation; pointwise stationary; fluid models; abandonments; stochastic networks
History: Received: August 1, 2005; accepted: July 20, 2007. Published online in *Articles in Advance* January 4, 2008.

1. Introduction

In this paper we formulate a general model of a stochastic processing network, a central feature of which is that exogenous arrival rates are allowed to vary both temporally and stochastically. As in the antecedent work cited below, we assume that each of the server pools that provide the system's processing capacity has many servers. In the context of that model, we study two interrelated problems: capacity choice and dynamic system control. This class of models substantially generalizes our earlier work on the interrelated problems of staffing and routing in telephone call centers (Harrison and Zeevi 2005; Bassamboo et al. 2006a, b) by allowing multiple processing operations for individual jobs or customers, and the notion of *differentiated processing modes* that will be illustrated in §5.

Adopting essentially the same framework as in Harrison (2003), we conceptualize a stochastic processing network in terms of resources, activities, and units of flow; the historical antecedents of that conceptualization are discussed in Harrison (2002). In addition to its greater flexibility for representing physical processing capabilities, the network model considered here has a more general economic structure than the call center models analyzed in Harrison and Zeevi (2005) and Bassamboo et al. (2006a, b). These features will be explained in §2, after some formal definitions have been introduced.

We are concerned with a two-stage decision problem: At the higher level, the system manager must choose a vector $b = (b_1, \ldots, b_r)$ the components of which may be loosely described as the capacities of various processing centers. At the lower level, given b, the system manager must choose a dynamic policy for allocating capacity to processing activities over a fixed and finite time interval. By assumption, the capacity vector b remains fixed over that time interval, but our analytical framework can easily be adapted to settings where *b* is reset periodically, subject to certain realistic constraints (see §8).

We propose in this paper a tractable modeling framework for the two-level problem of design and control. The proposed framework substitutes a seemingly crude PSFM for the finely structured stochastic processing network allowing development of practical policies with a computational burden that is manageable for systems of realistic size. The term PSFM reflects a blend of two concepts: fluid models and pointwise stationarity. Fluid models are macroscopic approximations of the original system that suppress low-level stochasticity, substituting mean flow rates for stochastic primitives. The rigorous justification of such models is based on the functional law of large numbers, the application of which is often referred to as "taking fluid limits." (Examples of recent work dealing with fluid analysis of nonstationary queuing systems include Mandelbaum et al. 1998 and Whitt 2006.) The term *pointwise stationary approximation* was introduced in Green and Kolesar (1991) in the context of a simple Markovian queuing model with nonstationary arrivals. It reflects the idea of using steady-state analysis at each instant in time while "freezing" the arrival rate. This idea was made rigorous in Whitt (1991); for further refinements see Massey and Whitt (1998). The recent publication Green et al. (2007) surveys work on setting staffing levels in service systems to meet time-varying demand.

The PSFM and the method we develop for its analysis are direct generalizations of earlier work on call centers, where two types of evidence were advanced to justify the approach: Numerical examples were analyzed to show the accuracy of our proposed procedure on small but arguably representative systems. Limit theory was developed to show the "asymptotic optimality" of that procedure in a parameter regime of practical interest. Here there will be little mention of formal limit theory, but building on the intuition developed in Bassamboo et al. (2006a, b) and Harrison and Zeevi (2005), we shall describe conditions under which we believe our proposed approach is a good one. Our intention is to make a transition in this paper from the formal, foundational flavor of Bassamboo et al. (2006a, b) to a PSFM "calculus" that is broadly accessible, the primary emphasis being on modeling and practical computation.

The remainder of the paper is structured as follows. In §2 we formulate a conventional, "exact" model of the stochastic processing network described above. Section 3 specifies our approximating PSFM for that system. Adopting the approximate model thereafter, we formulate in §4 the design and control problems referred to above, then "solve" those problems (that is, explain how an optimal capacity plan and an optimal control strategy can be computed) and interpret the solution. Section 5 is devoted to an elaborate example of back-office processing operations, which serves to illustrate the multistage and the differentiated processing modes that are allowed in our current model formulation. Our goal is to demonstrate in a concrete setting the power and flexibility of our modeling framework, not to provide a worked-out numerical example. In fact, numerical values are provided for only some model parameters, and we end the discussion of the example before the PSFM model has been completely formulated. To further illustrate our PFSM-based solution procedure, §6 examines a variant of the simple network with discretionary routing that was originally studied by Wein (1991).

Section 7 is somewhat different from the rest of the paper. There we strive to show how substantive new modeling issues can be addressed by extending or modifying our PSFM framework. More specifically, we consider a problem of dynamic server allocation in a system where arriving jobs are given accurate estimates of their waiting times. This real-time "delay notification" will induce some jobs to balk rather than enter the queue, which ultimately has a favorable effect on system performance. We first prove a preliminary result that establishes consistency of a suitable waiting time estimate (that is, an estimate that properly anticipates the balking behavior of future arrivals). We then show that a complete solution to the problem is still obtainable in our augmented PSFM framework, although real-time delay notification represents a substantial escalation in model complexity. Section 8 discusses various extensions of our PSFM framework, and §9 recapitulates the assumptions about model parameters that are needed to justify a PSFM.

2. A General Processing Network Model

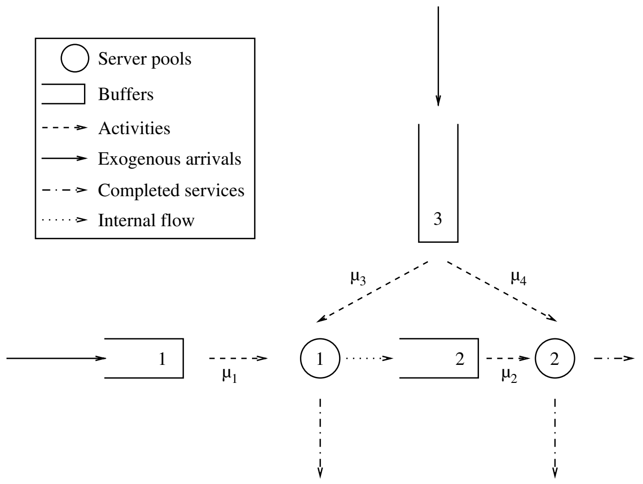
We consider a system comprised of r *server pools*, the kth of which contains b_k identical servers working in parallel. A rigorous justification of the approximation proposed in this paper requires that $b_1, b_2, \ldots, b_r$ all be large, but in numerical examples studied to date, good results have been obtained with as few as 5–10 servers per pool. A server might represent, for example, an agent in a call center, a computer in a Web server farm, or a machine in a manufacturing context. (To avoid overly narrow interpretations, one could use the term *processing resource* instead of *server*. We prefer the latter term because it is shorter and more vivid.) By assumption, discrete units called *jobs* arrive exogenously from outside the system; jobs eventually leave when their processing is complete.

Another essential concept in our modeling framework is that of a processing *activity*: Servers engage in activities to transform jobs or to remove them from the system under study. There is stochastic variability associated with the exogenous arrival process and endogenous processing activities.

In the model studied here, as in Harrison (2003), jobs are divided into m different *classes*. Each job arrives with a class designation that summarizes what is known initially about its processing requirements (and may carry other information about the job as well). Each of the activities available to the system manager involves a server from some particular pool k processing jobs of some particular class i. The activity produces a transition event after a random amount of time. That transition event is called a *service completion*, and the random amount of time is the *service time*. The service completion may result either in the job's departure or in its transition from class i to another class i'. (In the latter case, we say that the activity "creates" a job of class i' as output.) The probabilities of these different transitions are specific to the activity: It might be possible for a different server pool to process class i jobs or for servers from pool k to process class i jobs in a different mode (see §5); in either of those cases the transition probabilities when the service completion occurs could be different.

As an example, Figure 1 portrays a stochastic processing network that was introduced in Wein (1991).

Figure 1 **Schematic Representation of a Network with Three Job Classes ($m = 3$), Two Server Pools ($r = 2$), and Four Activities ($n = 4$)**



Here jobs of class 1 arrive exogenously and are processed by servers from pool 1 (this is designated as activity 1), after which they become class 2 jobs and are eventually processed by servers from pool 2 (this is designated as activity 2). Jobs of class 3 also arrive exogenously, and they may be processed either by servers from pool 1 (this is designated as activity 3) or by servers from pool 2 (this is designated as activity 4). After service is completed, they depart the system.

We denote by n the total number of processing activities available to the system manager. For each $j = 1, \ldots, n$ we denote the job class being processed in activity j by $i(j)$, the server pool doing the processing by $k(j)$, and the probability that activity j "creates" a class l job as output by p_{lj}. In system Equation (6), readers will see how the "switching probabilities" p_{lj} influence the random evolution of our controlled processing network; further comments will be provided immediately thereafter. It is the jobs created by processing activities that constitute the "internal flows" referred to in Figure 1. We assume that the service times associated with activity j are exponentially distributed with rate μ_j and that those service times are independent of arrival processes and of one another.

Let R and A be an $m \times n$ matrix and an $r \times n$ matrix, respectively, defined as follows: for each $j = 1, \ldots, n$ set $R_{ij} = \mu_j$ if $i = i(j)$ and $R_{ij} = -p_{ij}\mu_j$ if $i \neq i(j)$; set $A_{kj} = 1$ if $k = k(j)$ and $A_{kj} = 0$ otherwise. Thus, one interprets R as an *input-output matrix*: Its (i, j)th element specifies the average rate at which activity j processes jobs of class i; a negative (i, j)th element is interpreted to mean that activity i "produces" jobs of class i. The matrix A is a *capacity consumption matrix*: Its (k, j)th element is 1 if activity j draws on the capacity of server pool k and is zero otherwise. We define an $m \times n$ matrix B by setting $B_{ij} = 1$ if $i(j) = i$ and $B_{ij} = 0$ otherwise; thus, B is an incidence matrix the elements of which indicate which job classes are processed by the various activities. For the system in Figure 1 we have

$$R = \begin{bmatrix} \mu_1 & 0 & 0 & 0 \\ -\mu_1 & \mu_2 & 0 & 0 \\ 0 & 0 & \mu_3 & \mu_4 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \tag{1}$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

2.1. Blocking and Reneging

As stated, jobs of various classes arrive exogenously and randomly over time. The term "exogenous" is used to distinguish these arrivals from the internal flows of newly created jobs described below. Newly arrived and newly created jobs may be "accepted" or "blocked"; jobs that are blocked leave the system immediately and have no influence on future arrivals or other future events. (For an alternative treatment, see de Véricourt and Zhou 2005.) Those jobs that are accepted but that cannot be served immediately are stored in infinite-capacity buffers dedicated to their job class and are then processed in a first-in-first-out manner within the job class when the server becomes available. These jobs may renege if forced to wait too long before the commencement of service.

To represent reneging in our formal mathematical model, each class i job is endowed with an exponentially distributed "impatience" random variable τ with finite mean $1/\gamma_i$, independent of the impatience random variables of other jobs, and of service times and arrival processes. A job reneges when its waiting time in the class-specific buffer (exclusive of its own service time) exceeds τ time units. Let $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_m)$ denote the *reneging rate matrix*.

To repeat, we assume in our initial model formulation that *any* job can be blocked at the time of its arrival or creation; however, there are penalties for such blocking. The process of deciding which jobs to accept and which to block is often called "admission control" or "input control" in the stochastic systems literature, and those terms will be used occasionally as synonyms for blocking in this paper. Of course, the assumption that any job can be blocked, either when first arriving in the system or after changing classes, is a strong one, but it is not crucial. As explained in §8, our analysis and conclusions remain valid when admission control is impossible for some job classes, provided that jobs of those classes renege at a fast enough rate. Conversely, under mild assumptions on the cost structure, the reneging rate may be zero for some classes whose arrivals *can* be rejected.

2.2. Poisson-Exponential Assumptions

In the initial specification of our network model, we assume exponential service, reneging time distributions, and doubly stochastic Poisson arrival processes. As discussed in §3, the Poisson-exponential assumptions are not essential for our analytical framework, but they simplify its exposition. (However, rigorous limit theory that supports the PSFM formulation has only been derived to date under these simplifying assumptions.)

The assumption of doubly stochastic Poisson arrivals, which played a central role in our previous work on telephone call centers (Harrison and Zeevi 2005, Bassamboo et al. 2006a, b), means the following: At any given time, new arrivals into classes $1, \ldots, m$ occur according to independent Poisson processes with intensity parameters $\lambda_1, \ldots, \lambda_m$, but the vector $\lambda = (\lambda_1, \ldots, \lambda_m)$ of instantaneous arrival rates is itself evolving as a stochastic process with arbitrary distribution. Although this very general representation of system inputs, allowing average arrival rates to vary both temporally and stochastically, is necessary for realism in many application contexts, it makes exact analysis virtually impossible.

2.3. Probabilistic Structure of the Exogenous Arrival Process

We consider a complete probability space $(\Omega, \mathscr{H}, \mathbb{P})$ on which are defined $2m + mn$ mutually independent, unit rate Poisson processes denoted by $N_i^{(\ell)} = (N_i^{(\ell)}(t)\colon 0 \le t < \infty)$ for $i = 1, \ldots, m$ and $\ell = 1, 2$, and $N_{ij}^{(3)} = (N_{ij}^{(3)}(t)\colon 0 \le t < \infty)$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The $2m$ Poisson processes corresponding to $\ell = 1, 2$ will be used to construct the arrivals and reneging jobs for each customer class, and the mn Poisson processes corresponding to $\ell = 3$ will be used to construct service completions and internal flows associated with the various processing activities.

On the same space there are defined m continuous, nonnegative arrival rate processes $\Lambda_i = (\Lambda_i(t)\colon 0 \le t \le T)$ satisfying $\mathbb{E}[\int_0^T \Lambda_i(t)\,dt] < \infty$ for $i = 1, \ldots, m$, independent of the Poisson processes $N_i^{(\ell)}$. Here, T represents the planning horizon. Of course, there may be job classes i for which $\Lambda_i \equiv 0$; we denote by $\mathscr{E}$ (mnemonic for *exogenous* arrivals) the complementary set of classes $i \in \{1, \ldots, m\}$ such that $\mathbb{E}\{\int_0^T \Lambda_i(t)\,dt\} > 0$. Now define

$$F_i(t) = N_i^{(1)}\left(\int_0^t \Lambda_i(s)\,ds\right), \quad i = 1, \ldots, m, \tag{2}$$

interpreting $F_i(t)$ as the cumulative number of class i arrivals up to time t. This is a standard construction of a doubly stochastic Poisson process (cf. Bremaud 1981). Put $F = (F(t)\colon 0 \le t \le T)$ where $F(t) = (F_1(t), \ldots, F_m(t))$. The construction of reneging jobs and completed services under a given control will be done in an analogous manner using the Poisson processes $N^{(2)}$ and $N^{(3)}$; see (6).

2.4. Control Formulation

The system manager confronts a two-stage decision problem. First, choose a *capacity vector* $b = (b_1, \ldots, b_r)$ in $\mathbb{R}_+^r$, whose kth component is the number of servers to be used in station k during the specified planning period. This decision will be referred to as a *capacity choice* or *capacity plan* interchangeably. By assumption it cannot be revised, as actual arrivals are observed during the period.

Second, the system manager chooses a dynamic control for workflow management, which has two components: admission control and server allocation. Mathematically, a *dynamic control* is defined as a pair of stochastic processes (X, U), where $U = (U(t)\colon 0 \le t \le T)$ takes values in $\mathbb{R}_+^m$ and has sample paths that are nondecreasing and right continuous with left limits, and $X = (X(t)\colon 0 \le t \le T)$ takes values in $\mathbb{R}_+^n$ and has sample paths that are Lebesgue integrable and right continuous with left limits. Writing $U(t) = (U_1(t), \ldots, U_m(t))$ for the *admission control* and $X(t) = (X_1(t), \ldots, X_n(t))$ for the *server allocation policy*, we interpret $U_i(t)$ as the cumulative number of class i jobs that are blocked up until time t and $X_j(t)$ as the number of servers engaged in activity j at time t.

The reader should note that integrality constraints are relaxed in our formulation of capacity choice and dynamic control. The approximating framework advanced in §3 is only applicable if the nonzero components of b are large, so the distinction between integer and noninteger valued decision variables is not significant; see further discussion in §3.

A control (X, U) is said to be admissible if it is nonanticipating in a suitable sense (see below) and there exist processes Z and Q, both taking values in $\mathbb{R}_+^m$, both having time domain $[0, T]$, and both necessarily unique, that jointly satisfy conditions (3)–(6) for $0 \le t \le T$. We interpret $Z_i(t)$ as the number of class i jobs in the system at time t and $Q_i(t)$ as the number of class i jobs that are waiting for service at time t. We call Z and Q the *jobcount* process and *queue length* process, respectively. The relationships that (X, U, Z, Q) must jointly satisfy for all $t \in [0, T]$ are the following:

$$U_i(t) - U_i(s) \le F_i(t) - F_i(s) + \sum_{\{j\colon i(j) \neq i\}} N_{ij}^{(3)}\left(\int_s^t \mu_j p_{ij} X_j(s)\,ds\right) \quad \text{for all } s \in [0, t), \tag{3}$$

$$AX(t) \le b, \tag{4}$$

$$Q(t) = Z(t) - BX(t) \ge 0, \tag{5}$$

$$\begin{aligned} Z_i(t) = {} & F_i(t) - N_i^{(2)}\left(\int_0^t \gamma_i Q_i(s)\right) \\ & - \sum_{\{j\colon i(j)=i\}} \sum_{\{i'\colon i' \neq i\}} N_{i'j}^{(3)}\left(\int_0^t \mu_j p_{i'j} X_j(s)\,ds\right) \\ & + \sum_{\{j\colon i(j) \neq i\}} N_{ij}^{(3)}\left(\int_0^t \mu_j p_{ij} X_j(s)\,ds\right) - U_i(t) \quad \text{for } i = 1, \ldots, m. \end{aligned} \tag{6}$$

Condition (6) is the system dynamics equation: The first term on the right-hand side represents exogenous arrivals of class i jobs; the second term represents reneging by class i jobs; the third term represents service completions for class i jobs; the fourth term represents the creation of class i jobs by endogenous activities; and the last term represents blockage of class i arrivals.

Because the service requirements and impatience random variables are exponentially distributed, we can express all the terms in (6) as time changes of unit rate Poisson processes. In particular, the instantaneous service rate associated with activity j and reneging rate for class i are $\mu_j X_j$ and $\gamma_i Q_i$, respectively. Further, activities in the set $\{j: i(j) = i\}$ "consume" class i jobs, and activities in the set $\{j: i(j) \neq i\}$ may "produce" class i jobs depending on the probabilities p_{ij}. As mentioned above, an admissible control is nonanticipating with respect to the filtration generated by prior arrivals, services, and abandonments, that is, the minimal filtration generated by the terms on the right-hand side of the system dynamics equation (6), cf. of Bremaud (1981, pp. 196–197).

From this restriction to nonanticipative controls, plus familiar properties of the Poisson process, one sees that (6) is consistent with the interpretation of the parameters p_{lj} given earlier. To be specific, consider an activity j, a job class i such that $i = i(j)$, and another job class $l \neq i$. Now suppose that the processing of a class i job is undertaken by means of activity j. The associated service time is exponentially distributed with mean j, and p_{lj} is the probability that the job makes a transition to class l when its service is complete, independent of all previous events.

The first admissibility constraint (3) requires that the number of blocked jobs be no greater than the number of arrivals (because of both exogenous arrivals and internal flows) during any time interval for each class. Constraint (4) requires that the number of servers in a given pool who are engaged in processing activities at a given time not exceed the total number of servers available in that pool. In our third constraint, (5), the ith component of the vector $BX(t)$ represents the total number of class i jobs being processed at time t, and (5) requires that this not exceed the jobcount for class i at time t. Given a control (X, U), one can view the jobcount process Z and the queue length process Q as the unique solution of (5) and (6), which can be constructed jump-to-jump starting from time zero. Because the primitive processes $N^{(\ell)}$ are independent Poisson processes, the probability of simultaneous jumps is zero; hence, there almost surely exists a pair (Z, Q) satisfying the aforementioned relationship.

2.5. Economic Data

The system manager's economic objective will be discussed further in §4; here we list the economic data and introduce some terminology. As in Bassamboo et al. (2006b), we assume that a penalty $p_i^a \geq 0$ is incurred each time a class i job reneges (the letter a is mnemonic for "abandonment"), a penalty $p_i^b \geq 0$ is incurred each time a class i arrival is blocked, and holding costs are continuously incurred at rate $h_i \geq 0$ (expressed in dollars per time unit, for example) for each class i job in the system. Also, we are given a cost $c_k > 0$ to use a server in pool k over the entire planning horizon $(k = 1, \ldots, r)$.

The new economic element referred to in §1 is the following: We assume that each activity j generates *variable cost* at a (possibly negative) rate of v_j per time unit. If an activity j produces a lump-sum cost of expected magnitude δ_j on completion of the associated service, we represent this in our framework by setting $v_j = \mu_j \delta_j$, where μ_j is the average service rate (the reciprocal of the mean service time) for activity j. The value of this added model feature is illustrated in §7.

2.6. Structural Restrictions of the Model

Compared with the general formulation of a "stochastic processing network" laid out in Harrison (2003), the model described in this section is restrictive in two ways. First, we assume that each processing activity involves a single server from a specified pool, whereas simultaneous resource requirements were allowed in Harrison (2003). Second, in the current formulation, the units of flow are discrete "jobs," each activity involves a single job from a specified class as input, and the activity produces at most one job as output; one naturally thinks of the output job, if there is one, as a relabeling of the input job to reflect a change in its status or condition.

In contrast, the framework developed in Harrison (2003) allowed materials flowing through the system to be either discrete or continuous and allowed activities with multiple inputs, multiple outputs, or both. As explained in §8, the restrictions highlighted in this paragraph are not essential for the analytical framework we develop, but they are natural for the applications we have in mind, and they simplify the exposition.

3. The PSFM

The system dynamics under a control (X, U), as described in (6), are not analytically tractable for either performance evaluation or control purposes. In this section we describe the PSFM of the stochastic processing network described in §2. It is vastly simpler than the original stochastic model and supports tractable analysis.

To better understand the nature of this PSFM, first consider the fluid analog of the original system model, which is obtained by replacing all Poisson streams by fluid flows at their respective rates. Using this substitution, one gets the following system dynamics:

$$Z_i(t)=\int_0^t \Lambda_i(s)\,ds-\int_0^t (RX)_i(s)\,ds-\int_0^t \gamma_i Q_i(s)\,ds-U_i(t) \quad \text{for } i=1,\ldots,m, \tag{7}$$

for all $t\in[0,T]$. Further assume that the process U can be expressed in integral form; that is, for all $i=1,\ldots,m$ there exists a nonnegative real-valued process Y_i such that for all $t\in[0,T]$,

$$U_i(t)=\int_0^t Y_i(s)\,ds. \tag{8}$$

We interpret $Y_i(t)$ as the rate at which the jobs of class i are blocked at time t. One can then express the system dynamics informally in the following differential form:

$$\frac{dZ(t)}{dt}=\Lambda(t)-RX(t)-\Gamma Q(t)-Y(t), \tag{9}$$

for all $t\in[0,T]$, where $Y(t)=(Y_1(t),\ldots,Y_n(t))$. This fluid approximation is based on the functional strong law of large numbers and hence provides an accurate approximation to the original stochastic system when the flow of work through the system is "large." While this approximation eliminates "lower-order" stochastic fluctuations, it still leaves a complicated control problem, because the system dynamics are given by a differential equation with "random driver" $\Lambda(\cdot)$.

The main idea underlying the proposed PSFM is to further simplify (9) by eliminating transient dynamics. Imagine that at time $t\in[0,T]$ we "freeze" in the system all time-dependent quantities, except the job-count and queue length variables, whose dynamics are described by (9). In particular, for all times $s\ge t$ we fix $\Lambda(t)=\lambda$, $X(t)=x$, and $Y(t)=y$. Further, let $q_t(s)$ and $z_t(s)$ denote the queue length and jobcount vector. Then one has

$$\frac{dz_t(s)}{ds}=\lambda-Rx-\Gamma q_t(s)-y \tag{10}$$

for $s\ge t$. Further, $(q_t(s), z_t(s))$ satisfies $z_t(s)=q_t(s)+Bx$ for all $s\ge t$. The subscript t captures the dependence of these quantities on the values that have been frozen at time t. For any initial condition $(z_t(t), q_t(t))\in[0,\infty)\times[0,\infty)$, it is straightforward to verify that the queue length and headcount converge exponentially fast with a characteristic time constant proportional to the reneging rate. In particular, if we take an infinite time horizon, then

$$(z_t(s), q_t(s))\to(\bar z_t, \bar q_t) \quad \text{as } s\to\infty, \tag{11}$$

where $(\bar z_t, \bar q_t)$ satisfy the following relationship:

$$\lambda=Rx+\Gamma\bar q_t+y \quad\text{and}\quad \bar z_t=\bar q_t+Bx. \tag{12}$$

To obtain the PSFM equivalent of the original processing network, we simply "unfreeze" (λ,x,y) in (12) and consider again the original time interval $[0,T]$. The dynamic evolution of the system under a control (X,Y) is then determined by the following instantaneous flow-balance equation:

$$\Lambda(t)=RX(t)+\Gamma Q(t)+Y(t), \tag{13}$$

for all $t\in[0,T]$. Hence, the PSFM is obtained by first disregarding routine stochastic fluctuations, considering a fluid model of the original system dynamics, and then compressing the evolution of these dynamics over infinite time to a single point $t\in[0,T]$. The latter compression explains use of the term *pointwise stationary*.

For the PSFM we define an admissible control as a pair of processes X and Y, taking values in $\mathbb{R}^n_+$ and $\mathbb{R}^m_+$, respectively, that satisfy

$$AX(t)\le b \tag{14}$$

and

$$RX(t)+Y(t)\le\Lambda(t) \tag{15}$$

for all $t \in [0, T]$. It is easy to see that the set of admissible controls is nonempty. We associate with an admissible control (X, Y) a triple of processes (U, Q, Z) via (8), (13), and

$$Z(t) = BX(t) + Q(t) \tag{16}$$

for $0 \le t \le T$. Here (13) serves to define the queue length process Q for the admissible control (X, Y). (Of course, it is essential in this regard that every class $i = 1, \ldots, m$ have a *strictly* positive reneging rate γ_i, as assumed in §2. However, the PSFM construction can be extended to models where $\gamma_i = 0$ for some classes i, provided that other conditions are satisfied; see §8 for further discussion of this matter.) U and Z have the same interpretation as in §2, namely, $U(t)$ is a vector whose coordinates represent the cumulative number of jobs blocked until time t, and Z is the jobcount process.

3.1. Discussion

When does the PSFM described above provide a "reasonable" approximation of the original system? The reduction from the original dynamics to (9) is based on a high-flow volume assumption, while the reduction from (9) to (13) is appropriate when the system exhibits high turnover rates, that is, arrival rates, service rates, and reneging rates are large. (Note that the convergence in (11) is "fast" when the reneging rates are large.) In this environment, the PSFM combines fluid-flow dynamics with negligible transient behavior. (For a more formal treatment and supporting limit theory, cf. Bassamboo et al. 2006a, b).

In terms of distributional assumptions, the stochastic model in §2 assumes that the arrival process is doubly stochastic Poisson and that the service times and impatience variables have an exponential distribution. The memoryless property of the exponential distribution allows us to express the system dynamics (6) as a simple time change of Poisson processes. When the service times and impatience random variables follow general distributions, the analysis in Whitt (2006) suggests an approximating PSFM the instantaneous flow balance equation of which analogous to (13), has the form

$$\Lambda_i(t) = (RX)_i(t) + f_i(Q_i(t)) + Y_i(t), \quad i = 1, \ldots, m, \tag{17}$$

where the input-output matrix R depends on mean service times (and not on other characteristics of the service time distributions), exactly as in §2, and $f_i(\cdot)$ is a function that specifies the abandonment rate for class i jobs. When the impatience distribution is exponential with parameter γ_i, then $f_i(q_i) = \gamma_i q_i$, and in the general case $f_i(\cdot)$ depends on the distribution of the impatience random variable. We believe that (17) can be justified as a rigorous approximation of the underlying stochastic system in the spirit of the limit theory developed in Bassamboo et al. (2006a, b); see Whitt (2006) for a fluid analysis of a single-server system with general "impatience" distribution and an explicit characterization of $f_i(\cdot)$ in that context. It is also reasonable to speculate that the PSFM would still be a valid approximation to the original system dynamics when the arrival process is more general than doubly stochastic Poisson, e.g., a point process with stochastic intensity (cf. Bremaud 1981).

4. Design and Control via PSFMs

In this section we state the economic objective of the system manager in the context of our original stochastic processing network. We then state the solution prescribed by the PSFM for the capacity planning problem and the dynamic control problem and afterward explain the logic that supports our prescription.

4.1. Economic Objective

Consider the stochastic processing network described in §2 and recall from §1 the meanings of the economic parameters p_i^a, p_i^b, h_i, v_j, and c_k. Given a planning horizon $T > 0$, the total cost associated with a capacity vector b and corresponding admissible control (X, U) is

$$\sum_{k=1}^{r} c_k b_k + \sum_{i=1}^{m}\biggl(p_i^b U_i(T) + \int_0^T h_i Q_i(s)\,ds + p_i^a N_i^{(3)}\biggl(\int_0^T \gamma_i Q_i(s)\,ds\biggr)\biggr) + \sum_{j=1}^{n}\int_0^T v_j X_j(s)\,ds. \tag{18}$$

The objective of the system manager is to choose a capacity vector b and an admissible control (X, U) that jointly minimize the expected total cost.

In this formulation, the reneging and blocking penalty terms can be viewed as "dualizing" constraints on the fraction of reneging jobs and fraction of blocked jobs, respectively, with the class-specific penalties p_i^a and p_i^b interpreted as Lagrange multipliers. (A rigorous justification of this statement will be given in a separate paper.)

4.2. PSFM-Based Design and Control Policies

Because the exact formulation presented above is not analytically tractable, we now recast the system manager's problem in the context of our approximating PSFM. Let $p = (p_1, \ldots, p_m)$ be defined via

$$p_i := \min\left(p_i^b, p_i^a + \frac{h_i}{\gamma_i}\right) \tag{19}$$

for all $i = 1, \ldots, m$. Elements of the vector p are referred to as *effective loss penalties*.

To understand the basis for that terminology, consider an operating environment in which, as a matter of policy, each newly arrived or newly created job is either assigned to an unoccupied server immediately or else never served at all. (In our approximating PSFM, an optimal policy has essentially this character, but the manager of a real system will typically not interpret this policy prescription literally. That is, a system manager has discretion about how a PSFM-based policy will be "translated" in a realistic operating environment; see §§4.3 and 4.4 of Bassamboo et al. 2006a, and §4.2 in Bassamboo et al. 2006b for further details and discussion of such an implementation.) Once the system manager has relegated a class i job to the latter category, meaning that the he or she has decided to "lose" the class i job rather than serve it, there is still a choice to be made between blocking the job and just waiting for it to abandon. If the blocking option is chosen, a penalty of p_i^b is incurred; if the abandonment option is chosen, the expected time required for the job to abandon is $1/\gamma_i$. Thus the expected total cost incurred is $p_i^a + h_i/\gamma_i$. Of course, the system manager will choose the less-expensive means of "losing" the customer. Thus, the expected cost per class i customer lost is p_i.

Now consider the following LP: Choose $x \in \mathbb{R}^n$ to

$$\begin{aligned} &\text{minimize} \quad p \cdot (\lambda - Rx) + v \cdot x \\ &\text{subject to} \quad Rx \le \lambda, \quad Ax \le b, \quad x \ge 0, \end{aligned} \tag{20}$$

where $v = (v_1, \ldots, v_n)$ and $x \cdot y$ represents the scalar product of vectors x and y. For $\lambda \in \mathbb{R}_+^m$ and $b \in \mathbb{R}_+^r$, let $x^* := \phi(\lambda, b)$ be an optimal solution of LP (20), and let $\pi(\lambda, b)$ be the optimal value of the LP. Thus ϕ: $\mathbb{R}_+^m \times \mathbb{R}_+^r \to \mathbb{R}_+^n$ maps the right-hand side of (20) to the solution set. Assuming that the exogenous arrival rates $\Lambda(\cdot)$ are observable, we propose the following PSFM-based solution. (In §8 we discuss relaxing the assumption that Λ is observable.)

4.2.1. Capacity Planning. Let b^* be the solution to the following stochastic programming problem: Choose $b \ge 0$ to

$$\text{minimize} \quad c \cdot b + \mathbb{E}\left[\int_0^T \pi(\Lambda(t), b)\, dt\right]. \tag{21}$$

It can be shown easily that under general conditions the objective function is convex and there is a finite minimizer b^* in (21); see Harrison and Zeevi (2005). As noted there, it is a simple exercise in Fubini's theorem to show that the optimization problem (21) can be recast as follows:

$$\text{minimize} \quad c \cdot b + T \int_{\mathbb{R}_+^m} \pi^*(\lambda, b)\, dF(\lambda), \tag{22}$$

where the cumulative distribution function F is given by

$$F(\lambda) := \frac{1}{T}\int_0^T \mathbb{P}\{\Lambda(t) \le \lambda\}\, dt \quad \text{for } \lambda \in \mathbb{R}_+^m. \tag{23}$$

One interprets $F(\lambda)$ as the expected fraction of time (within the planning period under study) during which $\Lambda(\cdot) \le \lambda$. Although the formulation that uses the distribution $F(\lambda)$ is perhaps not very useful operationally, it is conceptually illuminating. In particular, it is evident that the optimization problem articulated above is a *two-stage LP with recourse*: At the first stage a system manager chooses the capacity vector b and incurs cost $c \cdot b$; then a random demand vector λ with distribution F is observed, and, given that observation, the system manager chooses at the second stage a vector x of activity levels that solve the linear programming problem (20).

The particular kind of two-stage problem embodied in (22) is sometimes called a *multi-dimensional newsvendor problem*. With regard to numerical solution techniques, various exact methods can be used when the distribution F concentrates its mass on a relatively small number of points; in the more general case, methods based on Monte Carlo simulation can be used. (The reader is refereed to Harrison and Zeevi 2005 for further discussion of numerical solution methods.)

4.2.2. Dynamic Server Allocation. For each $t \in [0, T]$ set

$$X^*(t) = \phi(\Lambda(t), b^*). \tag{24}$$

For each time $t \in [0, T]$, given the arrival rate vector $\Lambda(t)$, the optimal control vector $X^*(t)$ tells how servers in each pool should be allocated to different processing activities. That is, roughly speaking, $X^*(t)$ partitions each server pool into subpools that are dedicated to jobs of particular classes, plus (possibly) a residual set of servers that are kept idle because there is no work for them to do at time t. If the number of class i jobs present at time t is greater than the total number of servers from different pools that are dedicated to class i, then the remaining class i jobs are to wait in queue for later service.

Because $X^*(t)$ depends on $\Lambda(t)$, which may change continuously, the partitioning of server pools may change continuously as well. Of course, a practical implementation of our PSFM-based prescription would have to make some compromise with this idealization. One might, for example, reassess Λ and recompute X^* at relatively short time intervals (15-minute intervals would be plausible in a call center context), reassigning servers in accordance with that calculation as they complete services that were under way at the review point.

4.2.3. Admission Control. Partition the job classes into two sets $\mathcal{S}_a$ and $\mathcal{S}_b$, defined as follows:

$$\mathcal{S}_a = \left\{ i \in \{1, \ldots, m\}: p_i = p_i^a + \frac{h_i}{\gamma_i} \right\}$$
$$\mathcal{S}_b = \{1, \ldots, m\} \backslash \mathcal{S}_a. \tag{25}$$

Then for each $i = 1, \ldots, m$ set

$$Y_i^*(t) = \begin{cases} \Lambda_i(t) - (RX^*(t))_i & \text{if } i \in \mathcal{S}_b, \\ 0 & \text{if } i \in \mathcal{S}_a. \end{cases} \tag{26}$$

The optimal admission control Y^* described by (26) does not block any jobs from classes belonging to the set $\mathcal{S}_a$. However, (26) implies that $Q_i \equiv 0$ for $i \in \mathcal{S}_b$, which means that jobs from those classes are to be blocked if they cannot be served immediately.

4.3. Supporting Logic

This section explains the logic underlying the PSFM-based prescriptions discussed. We start by formulating the PSFM approximation to (18). Substituting the integral representation (8) of the admission control U and defining m vectors p^a, p^b, and h in the obvious way, one finds that the total cost associated with capacity vector b and corresponding admissible PSFM control (X, Y) can be expressed in vector notation as

$$\mathcal{J}(b, X, Y) := c \cdot b + \left[\int_0^T (p^b \cdot Y(t) + h \cdot Q(t) + p^a \cdot \Gamma Q(t))\, dt \right] + \int_0^T v \cdot X(t)\, dt. \tag{27}$$

Now solving for $Q(t)$ in the PSFM flow-balance Equation (13), and substituting this in (27) gives

$$\mathcal{J}(b, X, Y) = c \cdot b + \int_0^T ((p^b - h^T \Gamma^{-1} - p^a) \cdot Y(t) + (h^T \Gamma^{-1} + p^a) \cdot (\Lambda(t) - RX(t)) + v \cdot X(t))\, dt. \tag{28}$$

In the PSFM, the system manager's problem can be stated in the following hierarchical form: First, choose $b \geq 0$ before Λ is observed; second, as Λ is observed, and given b, choose $X(t)$ at each time t to satisfy $X(t) \geq 0$, $AX(t) \leq b$, and $RX(t) \leq \Lambda(t)$; and finally, given b and X, choose $Y(t)$ at each time t to satisfy $0 \leq Y(t) \leq \Lambda(t) - RX(t)$.

Given both b and X, one sees that the integrand (that is, the instantaneous cost rate) in (28) is minimized at time t by setting $Y_i(t) = 0$ if $(p_i^b - p_i^a - h_i/\gamma_i) \geq 0$ and setting $Y_i(t) = (\Lambda(t) - RX(t))_i$ [the maximum value it can take based on the admissibility condition (15)] otherwise. This is the admission control specified in (26). The above analysis reduces the total cost to

$$\mathcal{J}(b, X, Y^*) = c \cdot b + \int_0^T (p \cdot (\Lambda(t) - RX(t)) - v \cdot X(t))\, dt, \tag{29}$$

where p is the effective loss penalty vector. It is easy to see that for any given capacity vector b the dynamic control $X^*(t) = \phi(\Lambda(t), b)$ minimizes the integrand in (29) for each time instant $t \in [0, T]$ and for every realization of the arrival rate Λ. Thus we have the following result.

Proposition 1. *For any given capacity vector b, let (X^*, Y^*) be the dynamic control defined by* (24) *and* (26) (*taking b as input*). *Then for any other admissible control (X, Y) we have*

$$\mathcal{J}(b, X^*, Y^*) \le \mathcal{J}(b, X, Y). \tag{30}$$

Substituting the optimal server allocation rule X^* in (29), we arrive at the stochastic programming problem (21). The next result summarizes the optimality of (b^*, X^*, Y^*).

Proposition 2. *Let b^* be the solution of the optimization problem* (21). *Then the capacity vector b^* and the dynamic control (X^*, Y^*) defined in* (24) *and* (26) (*taking b^* as input*) *jointly minimize $\mathbb{E}[\mathcal{J}(b, X, Y)]$, where $\mathcal{J}(\cdot,\cdot,\cdot)$ is given by* (27).

5. A Back-Office Processing Example

In a standard call center model there is at most one activity for a given i and k (that is, servers from a given pool can process jobs from a given class either in one way or else not at all), and all service completions result in job departures. To illustrate the power of our current, more general modeling framework, Tables 1, 2, and 3 describe a system for processing credit applications. New applications are described as either simple or complex based on certain superficial characteristics. Of the simple applications, 15% are eventually found to be tenuous, which means that the credit decision is not clear-cut and hence must be made by a senior agent. The other 85% of simple applications can be processed to completion by a junior agent. In contrast, 38% of the complex new applications are eventually found to be tenuous, and a junior agent can tackle a complex new application in either of two modes.

The first of those (activity 2) is to proceed with the intention of processing to completion. If the application turns out to be tenuous, then the junior agent will finish all "preprocessing tasks" (obtaining a credit report, filling in missing information on the application form, etc.) and then put it aside for later disposition by a senior agent. The second way a junior agent can tackle a complex new application (activity 3) is to simply undertake the pre-processing tasks and pass it along to a senior agent, which saves a good deal of wasted effort when the application turns out to be tenuous. Activities 2 and 3 illustrate the phenomenon of "differentiated processing modes" referred to in §1.

There is no need to explicitly include job class 0 in our mathematical model of this clerical operation. The natural system representation has $r = 2$ server pools, $m = 5$ job classes, and $n = 8$ processing activities. From Tables 1, 2, and 3 we derive the following input-output matrix R, resource consumption matrix A, and incidence matrix B (recall that the rows of R correspond to job classes, rows of A correspond to server pools, rows of B correspond to job classes, and columns of each matrix correspond to activities):

$$R = \begin{bmatrix} 1/32 & 0 & 0 & 1/21 & 0 & 0 & 0 & 0 \\ 0 & 1/57 & 1/21 & 0 & 1/40 & 0 & 0 & 0 \\ -1/213 & 0 & 0 & 0 & 0 & 1/28 & 0 & 0 \\ 0 & -/150 & 0 & 0 & 0 & 0 & 1/35 & 0 \\ 0 & 0 & -1/21 & 0 & 0 & 0 & 0 & 1/30 \end{bmatrix},$$

Table 1 Server Pools for Credit Agency Example

Pool number	Description of servers
1	Junior agents (limited training and experience)
2	Senior agents (fully trained and experienced)

Table 2 Job Classes for Credit Agency Example

Job class	Description
0	Fully processed applications (leaving the system)
1	Simple new applications
2	Complex new applications
3	Preprocessed simple applications known to be tenuous
4	Preprocessed complex applications known to be tenuous
5	Preprocessed complex applications not further classified

Table 3 Processing Activities for Credit Agency Example

Activity number	Server pool	Input class	Mean service time (mins)	Output class	Probability
1	1	1	32	0	0.85
				3	0.15
2	1	2	57	0	0.62
				4	0.38
3	1	2	21	5	1
4	2	1	19	0	1
5	2	2	40	0	1
6	2	3	28	0	1
7	2	4	35	0	1
8	2	5	30	0	1

and

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The next step in the model specification is to identify the reneging rate γ_i for each job class $i = 1, \ldots, 5$. In the context of telephone call centers, reneging is usually called *abandonment*, and it is natural to assume that only callers waiting in queues abandon. However, in our credit agency example, reneging might correspond to a potential customer withdrawing his or her application because credit has been secured elsewhere, in which case the potential for reneging continues while the job is being processed. That structural change is easy to accommodate in our analytical framework, but discussion of the matter will be postponed to §8. The ability to have different reneging rates for different job classes is potentially important in many contexts. Imagine, for example, that customers submitting credit applications might be notified when their applications pass certain initial hurdles and that such notifications reduce the customers' propensity to look elsewhere for credit.

The only remaining "physical flow" information required in modeling the credit agency example is a probabilistic description of the system inputs over a chosen time horizon $[0, T]$. As explained in §4, a striking feature of the PSFM formulation is that all necessary information about the input process is captured by a single probability distribution on $\mathbb{R}_+^m$. In the current example, because there are no exogenous arrivals of job class 3, this amounts to a single distribution on $\mathbb{R}_+^2$.

Regarding economic data, one must first specify the penalties p_i^a and p_i^b associated with reneging and blocking of each job class i (reflecting the expected present value of current and future lost business). It may be appropriate to stipulate positive holding costs h_i *in addition* to reneging and blocking penalties, if processing delays are thought to have a negative influence on future business, or simply to reflect the decrease in present value when the receipt of revenue is retarded by processing delays.

Recall from §2 that our general model formulation also includes activity-specific variable cost rates v_j. The power of that model feature is well illustrated by the current example: It may be that junior agents are more likely than senior agents to make mistakes in processing complex applications, and those mistakes eventually lead to higher default rates for the applications processed by juniors. The data presented in Tables 1, 2, and 3 do not reflect such quality considerations, but they can be accounted for in the variable cost rates associated with different activities.

6. Dynamic Control of a Network with Discretionary Routing

In this section we apply the PSFM framework to solve a dynamic control problem associated with the simple network pictured in Figure 1. Given reneging penalties p_i^a, blocking penalties p_i^b, and holding cost rates h_i for each class i ($i = 1, 2, 3$), we define the effective loss penalties p_i via (19). We assume throughout that

$$p_1 < p_2. \tag{31}$$

This means that it is economically preferable to lose a class 1 job (either by blocking the job or by allowing it to renege, depending on which of those modes is less costly) than to lose it later as a class 2 job. Condition (31) would be expected in virtually any application context. Another assumption on the cost structure is that $v_1 = 0, \ldots, v_4 = 0$. That is, there are no additional variable costs associated with processing activities. Thus, the last term in the total cost expression (18) is absent, as is the term $v \cdot x$ in the objective function of the LP (20) that is associated with our approximating PSFM.

We assume throughout this section that the capacity vector $b = (b_1, b_2)$ is fixed, and we address the system manager's dynamic control problem using the PSFM framework explained earlier. As in any example where variable activity costs are absent, the system manager's control problem in the PSFM context amounts to deciding, at each time t, given the observed vector $\Lambda(t)$, which job flows to "give up on."

That is, given that $\Lambda(t) = \lambda$, components of the nonnegative vector $\lambda - Rx$ appearing in the objective function of the LP (20) represent flow loss rates for the various job classes, and the objective at time t is to minimize the instantaneous cost rate associated with such losses. If there exists an $x \geq 0$ such that $Ax \leq b$ and $Rx = \lambda$, meaning that all exogenous arrivals can be processed to completion without losses, then it is optimal to take $X(t) = x$ when $\Lambda(t) = \lambda$, and the corresponding minimum instantaneous cost rate is $\pi(\lambda, b) = 0$.

Given our economic assumption (31), and given the special structure of the example under discussion, it is obvious (and easy to prove rigorously) that in solving the LP (20) one can restrict attention to server allocation vectors x satisfying $\mu_1 x_1 = \mu_2 x_2$, or equivalently,

$$x_2 = \alpha x_1 \quad \text{where } \alpha = \mu_1/\mu_2. \tag{32}$$

That is, one need only consider x vectors, which process class 1 jobs and class 2 jobs at the same rate, thus ensuring that no class 2 jobs are lost. Of course, we can use (32) to eliminate x_2 from the LP formulation (20). Doing so symmetrizes the roles of server pool 1 and server pool 2, as follows: In the reduced system model that one gets by using (32) to eliminate x_2, there is one activity that processes class 1 jobs, creates no new jobs, and consumes capacity from both server pools, and there are two activities that process class 3 jobs, create no new jobs, and consume the capacity of either one server pool or the other.

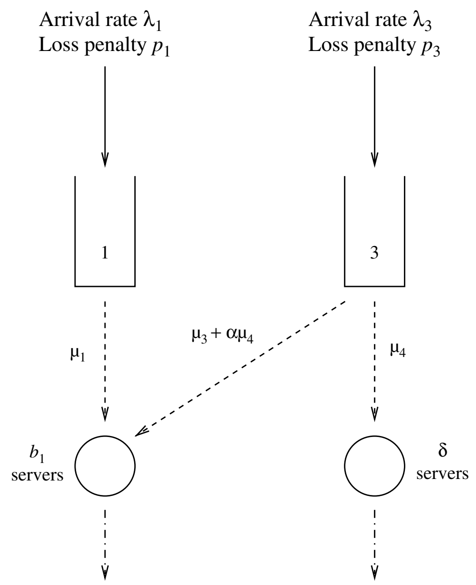
To further reduce the system manager's problem, let us assume that server pool 1 is the limiting factor in the processing of class 1 jobs. That is, let us assume $\mu_1 b_1 \leq \mu_2 b_2$, or equivalently,

$$\delta := b_2 - \alpha b_1 \geq 0. \tag{33}$$

(Because of the symmetry noted here, a precisely analogous reduction applies in the case where $\mu_1 b_1 > \mu_2 b_2$, as readers can easily verify.) Now let us fix for the moment a feasible value for x_1 in the LP (20), meaning that $0 \leq x_1 \leq \lambda_1$ and $x_1 \leq b_1$. This choice commits x_1 servers from pool 1 and αx_1 servers from pool 2 to the processing of class 1 arrivals, so one sees from Figure 1 that the maximum rate at which class 3 arrivals could be processed is

$$\mu_3(b_1 - x_1) + \mu_4(b_2 - \alpha x_1) = (\mu_3 + \alpha\mu_4)(b_1 - x_1) + \mu_4 \delta.$$

Figure 2 **An Equivalent Reduced System**



Let us denote by $u_3(x_1)$ this upper bound on the processing rate for class 3. Viewed as a function of x_1, the lowest achievable loss rate for class 3 is $[\lambda_3 - u_3(x_1)]^+$, and one can thus state the system manager's optimization problem (20) as follows: Choose the single decision variable x_1 to

$$\begin{aligned} &\text{minimize } p_1(\lambda_1 - \mu_1 x_1) + p_3[\lambda_3 - (\mu_3 + \alpha\mu_4) \\ &\qquad\qquad \cdot (b_1 - x_1) - \mu_4 \delta]^+ \\ &\text{subject to } \mu_1 x_1 \leq \lambda_1, \quad x_1 \leq b_1 \quad \text{and} \quad x_1 \geq 0. \end{aligned} \tag{34}$$

Now optimization problem (34) is equivalent to the problem of minimizing the instantaneous cost rate in the reduced system pictured in Figure 2, which can be solved by inspection: first allocate $\delta \wedge (\lambda_3/\mu_4)$ servers from the right-hand pool to process class 3 jobs. If there are still class 3 arrivals left over, allocate the capacity of the left-hand pool via the obvious greedy algorithm, giving class 1 priority if $p_1\mu_1 \geq p_3(\mu_3 + \alpha\mu_4)$ and class 3 priority otherwise. Each unit of the activity represented by the diagonal arrow in Figure 2 corresponds to allocating one server from pool 1 of our original system to activity 3 *and* allocating α servers from the original pool 2 to activity 4.

To translate this description into optimal values for the original decision variables $x_1, \ldots, x_4$, let us consider for concreteness the case where

$p_1\mu_1 \geq p_3(\mu_3+\alpha\mu_4)$. (The complementary case where $p_1\mu_1 < p_3(\mu_3+\alpha\mu_4)$ is slightly more complicated, and its treatment is left as an exercise.) As stated, an optimal solution will then give priority to class 1 in allocating servers from the left-hand pool in Figure 2, which means that

$$x_1^* = x_2^*/\alpha = (\lambda_1/\mu_1) \wedge b_1.$$

Also, the system manager in Figure 2 will dedicate as many servers as possible from the right-hand pool to the processing of class 3 arrivals, which is expressed mathematically as

$$x_4^* = (\lambda_3/\mu_4) \wedge \delta.$$

Finally, if there are more class 3 arrivals to be processed, the system manager will allocate to class 3 as many of the still-uncommitted servers in the left-hand pool as possible, which means that

$$x_3^* = [(\lambda_3 - \mu_4 x_4^*)/(\mu_3+\alpha\mu_4)] \wedge (b_1 - x_1^*).$$

This completes the specification of the optimal server allocations in our original system model, given that $\Lambda(t)=\lambda$. It remains only to determine the optimal input control policy. Let us define the sets $\mathcal{S}_a$ and $\mathcal{S}_b$ as in (25). If $i \in \mathcal{S}_b$ $(i=1,2,3)$, then the general solution developed in §4 calls for blocking any newly arrived or newly created class i jobs that cannot be served immediately. On the other hand, if $i \in \mathcal{S}_a$ $(i=1,2,3)$, then the general solution dictates that class i jobs *never* be blocked, regardless of system status at the time of their arrival or creation. In our idealized PSFM, the identity $x_2^* = \alpha x_1^*$ ensures that servers will always be available in pool 2 for processing newly created class 2 jobs, so it makes no difference whether $2 \in \mathcal{S}_a$ or $2 \in \mathcal{S}_b$, but the general articulation of the optimal input control policy given immediately above is still valid.

7. Real-Time Delay Announcement in Call Centers

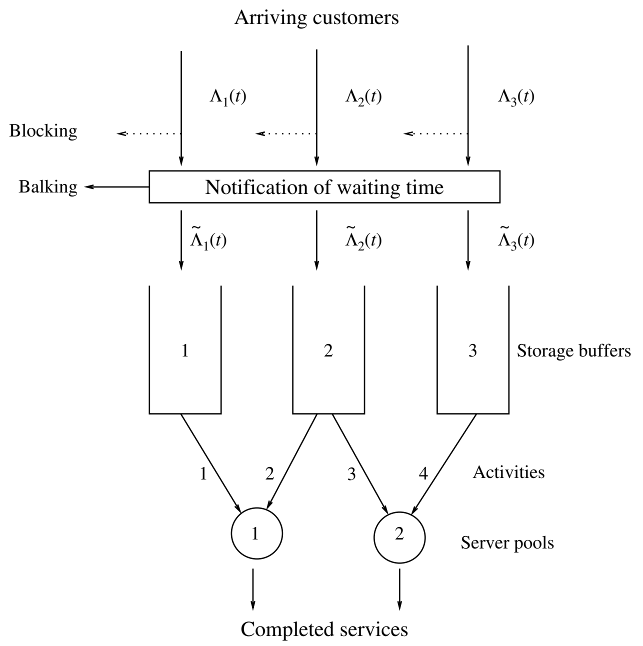
This section illustrates how the PSFM framework can be used to analyze the problem of dynamic control and capacity planning in a parallel server network where waiting times are announced to arriving customers. The motivating application is that of a multiclass, multiskilled telephone call center operation. In that context, arriving jobs are customers and servers are multiskilled agents (or customer service representatives) grouped into pools based on their common skill sets.

7.1. Problem Formulation

Consider the general stochastic processing network described in §2, but with no internal flows; that is, an arriving job leaves the system after completing service at one processing station. For such a network the input-output matrix R has nonnegative entries, these being the rates of service for the various combinations of customer class and server pool. An example of such a parallel server network is given in Figure 3.

Customers of various classes arrive to the network randomly over time. On arrival, customers are either blocked by means of a busy signal or admitted into the system. In the latter case they are given an estimate of their waiting time in the queue. (We assume that this estimate reflects the true anticipated waiting time in the system for a customer arriving at time t based on the queue length at that time; in particular, this quantity is not manipulated in any way by

Figure 3 **Schematic Representation of a Parallel Server Network that Announces Waiting Times to Arriving Customers**



Note. In this network there are three customer classes, two server pools, and four activities.

the system manager to affect customer expectations.) Blocked customers leave the system instantaneously (represented by the dotted horizontal lines in Figure 3). Based on the waiting time estimate, customers who are not blocked then decide whether to wait for service or balk. (Balking is represented by horizontal dashed lines emanating from the rectangular box.) Those customers who do not balk but who cannot be served immediately wait for their service in infinite-capacity buffers dedicated to their class. The system manager then routes these customers when agents become available, according to a server-allocation policy as described in §2.

We assume that each customer of class i is endowed with a random variable τ_i with cumulative distribution function G_i $(i = 1, \ldots, m)$, independent of all other stochastic primitives. This variable characterizes the customer's "willingness to wait," that is, the realized value of τ_i is the smallest announced delay that will cause a customer to balk. The role of τ_i is analogous to that of a "reservation price" that characterizes consumers in demand theory. (For technical reasons we assume that the distribution G_i has a density with finite support for all $i \in \{1, \ldots, m\}$; see the proofs of Propositions 3 and 4 for further details.)

A customer of class i admitted to the system at time $t \in [0, T]$ will *balk* if $\tau_i \leq \widetilde{W}_i(Q(t), X(t))$, where $\widetilde{W}_i(Q(t), X(t))$ is the waiting time estimate announced to class i customers arriving at time t, based on the queue length vector $Q(t)$ and the server allocation vector $X(t)$. Consequently, the *effective* arrival rate for class i at time t is given by

$$\tilde{\Lambda}_i(t) = (\Lambda_i(t) - Y_i(t))\bar{G}_i(\widetilde{W}_i(Q(t), X(t))),$$

where $\Lambda_i(t)$ is the nominal arrival rate into class i, $Y_i(t)$ is the blocking rate for class i, and $\bar{G}(\cdot) := 1 - G(\cdot)$.

The system manager must choose a capacity vector $b = (b_1, \ldots, b_r)$ whose components are the numbers of servers staffing each pool and a dynamic control (X, U) consisting of an admission control U and server allocation policy X, subject to the admissibility constraints in §2. (For simplicity, we ignore integrality constraints on the server allocations.) The total cost under a capacity vector b and dynamic control (X, U) is given by

$$\sum_{k=1}^{r} c_k b_k + \sum_{i=1}^{m}\left(p_i^b U_i(T) + \int_0^T h_i Q_i(s)\,ds\right), \tag{35}$$

where p_i^b and h are the blocking penalty and holding cost rate defined in §4. The system manager's objective is to minimize the expected total cost, given by the expectation of (35). This formulation can be easily modified to impose an economic penalty for customers who balk, but we have chosen not to pursue this extra layer of generality. (It is not difficult to see that if one introduces a linear cost of balking into the objective function, the main results derived in what follows can be easily modified to cover that setting.) For further work on the effects of announcing waiting time in service systems, see Armony and Maglaras (2004), Armony et al. (2005), Maglaras and Zeevi (2005), and Whitt (1999). The first three papers use fluid and diffusion approximations to make the problem tractable.

7.2. Waiting Time Estimates from Queue Lengths

We come now to the question of how a system manager should estimate the waiting time for class i fluid entering at time t. In the context of our approximating PSFM, the "right" answer is

$$\widetilde{W}_i(Q(t), X(t)) = Q_i(t)/(RX)_i(t), \tag{36}$$

as follows:

(a) The denominator $(RX)_i(t)$ on the right side of (36) is the instantaneous rate of fluid removal from buffer i at time t.

(b) As explained in §3, a key assumption underlying the PSFM is that $\Lambda(\cdot)$ changes slowly compared with the time that a quantum of fluid remains within the system. Thus the server allocation vector X under a good control policy also remains approximately constant over such a time span.

(c) The numerator $Q_i(t)$ on the right side of (36) is the amount of class i fluid that must be removed from the buffer before class i fluid arriving at time t can enter service.

7.3. The PSFM Formulation and Its Solution

Given a capacity vector b, we define an admissible control (X, Y) exactly as in §3: The key constraints are that $AX(t) \le b$ and $RX(t) + Y(t) \le \Lambda(t)$ for all $t \in [0, T]$, with the additional constraint that if for any time $t \in [0, T]$ and $i \in \{1, \dots, m\}$ $(RX)_i(t) = 0$, $Y_i(t) = \Lambda_i(t)$. The physical interpretation of these constraints is as follows: First, the system manager cannot assign servers to activities beyond the total number of servers available in each pool; second, the total rate of service and blocking in each customer class cannot exceed the arrival rate into that class; finally, if at any point in time the system manager decides not to serve a given customer class, then further arrivals into that class are blocked. (To facilitate the proof, we also assume that if for any $t \in [0, T]$ we have $(RX)_i(t) = 0$, then $Q_i(t) = 0$.)

Proposition 3. *For each admissible control* (X, Y) *there exists a unique nonnegative queue length process* Q *such that*

$$(\Lambda_i(t) - Y_i(t))\bar{G}_i(\widetilde{W}_i(Q(t), X(t))) = (RX(t))_i \quad \textit{for all } i = 1, \dots, m, \textit{ and } t \in [0, T]. \tag{37}$$

The left-hand side of (37) represents the net arrival rate for class i customers at time t, and the right-hand side represents the net output rate due to service completions. Thus (37) simply characterizes the stationary point for the system at each time instant, which is an obvious modification of the PSFM-based system dynamics (13) to account for balking. This formulation assumes that announcement of waiting time results in an effective arrival rate $\tilde{\Lambda}$ that is *consistent* with this information. In the PSFM the arrival rate translates "instantaneously" to queue lengths (see the discussion in §3).

In our PSFM formulation the system manager seeks admissible controls (X, Y) and a staffing vector b to

$$\text{minimize} \quad c \cdot b + \mathbb{E}\left[\int_0^T (p^b \cdot Y(t) + h \cdot Q(t))\,dt\right]. \tag{38}$$

Now given a staffing level b, as in §4, we can minimize the second term in (38) by minimizing the integrand for each point in time and for each realization of the arrival rate process Λ. For brevity, we define a function ψ: $\mathbb{R}^m_+ \times \mathbb{R}^n_+ \to [0, 1]^n$ as follows:

$$\psi_i(q, x) := \bar{G}_i\left(\frac{q_i}{(Rx)_i}\right). \tag{39}$$

Let $\Psi(q, x) = \operatorname{diag}(\psi_1(q, x), \dots, \psi_n(q, x))$. For $\lambda \in \mathbb{R}^m_+$ and $b \in \mathbb{R}^r_+$, let $\pi(\lambda, b)$ be the optimal value of the following optimization problem: choose $x \in \mathbb{R}^n_+$, $y \in \mathbb{R}^m_+$ and $q \in \mathbb{R}^m_+$ to

$$\begin{aligned} \text{minimize} \quad & h \cdot q + p^b \cdot y \\ \text{subject to} \quad & (\lambda - y)\Psi(q, x) = Rx, \\ & Ax \le b, \\ & x \ge 0, \quad q \ge 0, \quad y \ge 0. \end{aligned} \tag{40}$$

The first constraint in the optimization problem (40) is a restatement of (37) that serves to define the queue length vector q. The second constraint is a restatement of the admissibility condition (14). This optimization problem thus minimizes the cost rate in the PSFM at each time instant in $[0, T]$ subject to admissibility conditions on the chosen control.

Proposition 4. *For every* $\lambda \in \mathbb{R}^m_+$ *and* $b \in \mathbb{R}^r_+$, *there exists a finite valued solution* (x^*, y^*, q^*) *for the optimization problem* (40). *Further, there exists a measurable function* ϕ: $\mathbb{R}^m_+ \times \mathbb{R}^r_+ \to \mathbb{R}^n_+ \times \mathbb{R}^m_+ \times \mathbb{R}^m_+$ *such that* $\phi(\lambda, b)$ *solves* (40).

Put $(X^*(t), Y^*(t), Q^*(t)) = \phi(\Lambda(t), b)$ for all $t \in [0, T]$. The optimal controls are then given as follows: The system manager allocates $X^*(t)$ servers to each activity at each time $t \in [0, T]$ and blocks customers in class i when $Q_i(t) > Q^*_i(t)$.

The optimal capacity vector b^* is given by the following stochastic optimization problem: Choose $b \ge 0$ to

$$\text{minimize} \quad c \cdot b + \mathbb{E}\left[\int_0^T \pi(\Lambda(t), b)\,dt\right], \tag{41}$$

where $\pi(\lambda, b)$ is the value of the optimization problem (40), and the expectation operator is with respect to the distribution of the arrival rate Λ.

8. Discussion of the Model Assumptions

In this section we return to various issues that have been flagged as topics for further discussion in the body of the paper. Most of these involve potential generalizations of the network model laid out in §2 and of the PSFM approximation we have proposed for that model.

8.1. Cost Structure and Economic Objective

We have assumed that the holding cost rate for class i jobs at time t is a *linear* function of the queue length $Q_i(t)$. One reason for restricting attention to linear holding costs is the following: The heart of our PSFM formulation is the instantaneous flow balance Equation (13). The asymptotic theory developed earlier in Bassamboo et al. (2006a, b), only provides rigorous justification for an integrated form of (13). With a linear cost structure, the cumulative holding cost associated with class i jobs is proportional to the integral of $Q_i(\cdot)$, and hence the existing asymptotic theory suffices. A more delicate limit theory would be needed to rigorously justify the PSFM with nonlinear holding costs. On the other hand, for those who are prepared to accept our PSFM formulation without the support of a formal limit theory, the analysis in §4 can easily be modified to incorporate nonlinear holding cost functions.

8.2. Reneging Structure

As mentioned in §1, there are many applications (such as our credit agency example) where a job may renege while being processed. The PSFM framework can easily be modified to encompass such applications. For example, if the reneging rate is exactly the same for class i jobs being processed as for class i jobs in queue, the PSFM under an admissible control (X, Y) is defined by the following modification of the instantaneous flow-balance Equation (13):

$$\Lambda(t) = RX(t) + \Gamma Z(t) + Y(t).$$

The most general reneging structure that comes readily to mind is the following: There are reneging rates $\gamma_1, \ldots, \gamma_m$ associated with jobs waiting in buffers $1, \ldots, m$ (as in our original formulation) and reneging rates $\omega_1, \ldots, \omega_n$ associated with jobs being processed via activity $1, \ldots,$ activity n. This structure leads to the modified flow-balance equation

$$\Lambda(t) = RX(t) + \Gamma Q(t) + B\Omega X(t) + Y(t),$$

where $\Gamma = \operatorname{diag}(\gamma_1, \ldots, \gamma_m)$ and $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_n)$. The policy prescriptions developed in §4 extend to this more general setting without new complications.

8.3. Weaker Assumptions About Blocking and Reneging

Until now we have assumed that newly arrived and newly created jobs of *any* class can be blocked by the system manager and that *all* job classes renege at positive rates when queued. The former assumption serves only to simplify notation and exposition. A careful review of §§2–4 shows that some of the input control processes U_i can be forced to zero without materially changing any of our conclusions. Indeed, associating with class i jobs a sufficiently large blocking penalty p_i^b assures that those jobs are never blocked in our PSFM-based policy prescription, given that $\gamma_i > 0$ and that h_i and p_i^a are finite (see §4.2).

On the other hand, suppose there exists a job class i for which $\gamma_i = 0$ (no reneging). Assume that $h_i > 0$ and that newly arrived and newly created class i jobs can be blocked with penalty $p_i^b < \infty$. In the obvious way, we interpret (19) to mean that the effective loss rate for class i jobs is $p_i = p_i^b$ in this case, and (with one exception to be noted) the development in §§3 and 4 can proceed exactly as before. In particular, we have $i \in \mathcal{S}_b$, which means that newly arrived or newly created class i jobs are to be blocked if they cannot be served immediately.

The exception is the following: When $\gamma_i = 0$ for some classes i, it cannot be said that (13) serves to define the queue length vector $Q(t)$ under any control policy (X, Y) satisfying the admissibility conditions (14)–(15) because the diagonal matrix Γ is then singular. However, this indeterminacy in the PSFM formulation can be eliminated by adding the following complementarity condition:

$$\text{either } Q_i \equiv 0 \text{ or } Y_i \equiv 0 \text{ or both } (i = 1, \ldots, m). \qquad (42)$$

That is, the PSFM is well posed mathematically if one defines an admissible control as a triple (X, Y, Q), taking values in $\mathbb{R}_+^n \times \mathbb{R}_+^m \times \mathbb{R}_+^m$ and satisfying (13)–(15) plus (42). Condition (42) restricts attention from the outset to control policies that, for each job class, either exercise no input control at all or else block any new arrival into that class that cannot be served immediately. (Of course, this is a characteristic of the PSFM-based policy prescription in §4.2.)

In this discussion concerning a job class i for which $\gamma_i = 0$, we implicitly made the following assumption: The holding cost rate h_i is large enough, relative to the time scale on which $\Lambda(\cdot)$ evolves, that no rational system manager would hold class i jobs in buffer storage while waiting for the demand environment to change. If the holding cost rate *were* small enough to make such a strategy attractive, our PSFM formulation would not be appropriate. (See further comments in §9.)

8.4. Broadening the Definition of an Activity

As mentioned in §1, the model structure considered in this paper could be generalized to allow any or all of the following features: activities that consume continuous materials, rather than discrete "jobs," as inputs; activities that require multiple inputs, such as assembly operations; and activities that produce several outputs, such as refinery operations with by-products. These added features make construction of the "conventional" system model (see §2) substantially more complicated, but in terms of the PSFM eventually obtained, they manifest themselves in relatively simple ways: The input-output matrix R may have several positive entries in a given column to reflect multiple inputs and may have several negative entries in a given column to reflect multiple outputs. This generalization in the form of the R matrix does not affect the analysis in §4; the PSFM-based policy prescription and its interpretation are essentially unchanged.

Earlier work on stochastic processing networks (Harrison 2002, 2003) has also considered activities that consume the capacity of several different resources, such as industrial operations that involve both capital equipment and skilled labor, which leads to a capacity consumption matrix A that may have several positive elements in a given column. The construction and analysis of the PSFM that we have presented in §4 continue to make mathematical sense in this case. However, the interpretation offered there for the optimal control X^*, which involved dedicating servers to specific activities over short intervals of time, may be too simplistic, depending on the precise manner in which resources are employed.

8.5. Estimating Arrival Rates

Throughout this paper we have proceeded as if the arrival rate vector $\Lambda(t)$ were directly observable. However, what one actually observes in virtually all applications are individual arrivals, and then the "underlying arrival rates" must be estimated by some sort of averaging. In our approach, the estimation procedure obviously affects the PSFM-based resource allocation vectors $X^*(t)$ because $X^*(t)$ is computed from $\Lambda(t)$ via linear programming. In earlier work (Bassamboo et al. 2006a, b) we have described and justified dynamic control formulations that estimate arrival rates "on the fly."

The estimation of arrival rates is also important in determining our PSFM-based staffing vector b^* because the stochastic program (21) takes as input the distribution of the arrival rate process $\Lambda(\cdot)$. In Bassamboo and Zeevi (2006) we describe and rigorously justify a data-driven optimization scheme that computes b^* directly from arrival data. This constitutes an integrated approach to the estimation of $\Lambda(\cdot)$ and numerical solution of (21).

8.6. Frequent Adjustment of Capacity Levels

In §4 we have formulated the capacity choice problem with a fixed time horizon T, assuming that the capacity vector b, once chosen, must remain unchanged over $[0, T]$. In some large telephone call centers, where the "processing resources" are multiskilled human agents, work schedules can be staggered so as to change capacity levels every 30 minutes. To apply our method in such an environment one would undertake a separate analysis for each 30-minute segment of the working day.

For example, with a 12-hour work day one would solve 24 separate staffing-and-control problems, each with its own arrival rate data and each with a time horizon of $T = 30$ minutes. However, such an approach involves two important assumptions. The first is that capacity levels during various subintervals must be *specified in advance*, as opposed to dynamic capacity adjustment in response to observed demand. (This is a realistic restriction in many settings.) The second implicit assumption is that staffing costs associated with different subintervals are *separable*: There are no "smoothing costs" incurred when staff levels change abruptly from one hour to the next, nor do union contracts or personnel policies impose any restrictions on the staffing combinations that are available across the day. In reality, of course, scheduling constraints and smoothing costs *do* exist, and accounting for such considerations requires a more sophisticated version of our basic method. Roughly speaking, one needs to link the stochastic programs that set staffing levels for various subintervals within the work day, eventually solving a large, multistage stochastic optimization problem.

9. On the Conditions Justifying a PSFM

Expanding on the comments in §3, one may summarize the conditions needed to justify a PSFM formulation of the system manager's problem. First, the volume of work demands a large number of servers. Second, the arrival and service processes are "fast" compared with the rate of change in the demand rate Λ. Finally, even when capacity is insufficient to process all arriving demand, no job ever stays in the system long enough to see a significant change in $\Lambda(\cdot)$. This occurs when each job class either abandons at a relatively fast rate or is too expensive to store in the time scale at which changes occur in the demand environment. In the latter case a rational system manager will block such jobs when they cannot be processed immediately.

We conjecture that one may obtain a PSFM as a limit of the conventional network model described in §2. More specifically, the asymptotic parameter regime we discuss involves a fluid limit (where one increases all arrival rates by a large factor κ, increases the number of servers in each pool by that same factor κ, and rescales queue lengths and jobcount variables by κ as well) and a further acceleration of all arrival, service, and abandonment processes by a common large factor, which leads to "instantaneous equilibration" in response to any change in $\Lambda(\cdot)$. A rigorous asymptotic theory of this kind was developed in earlier work (Bassamboo et al. 2006a, b) for a less general class of processing system models.

Acknowledgments
This research was supported in part by National Science Foundation (NSF) Grant DMI-04-4765.

Appendix. Proofs

Proof of Proposition 1. The argument provided in the text of §4.3 is missing only technical details, which we omit in the interest of brevity. A complete proof is available from the authors upon request. □

Proof of Proposition 2. The proof follows from Proposition 1 by taking expectation of both sides and maximizing over the staffing level b. □

Proof of Proposition 3. Fix time $t \in [0, T]$ and recall that by assumption $(X(t), Y(t))$ satisfies $\Lambda(t) \geq RX(t) + Y(t)$. Fix $i \in \{1, \ldots, m\}$, and consider the equation $(\Lambda_i(t) - Y_i(t))\Psi_i(q_i) = (RX(t))_i$. Note that the left-hand side is nonincreasing and continuous in q and the right-hand side is a constant. The result follows from the fact that $\Psi(0) = 1$ and $\Psi(q_i) = 0$, where the function $\Psi(\cdot)$ is defined in (39). □

Proof of Proposition 4. Eliminating y from the optimization problem (40), we get the following optimization problem: Choose $x \in \mathbb{R}^n_+$ and $q \in \mathbb{R}^m_+$ to

$$\begin{aligned} \text{minimize} \quad & h\cdot q + p^b \cdot (\lambda - (Rx)\Psi^{-1}(q, x)) \\ \text{subject to} \quad & Rx \leq \lambda\Psi(q, x), \\ & Ax \leq b, \quad x \geq 0, \quad q \geq 0. \end{aligned} \tag{43}$$

Define the correspondence

$$\mathcal{D}(\lambda, b) := \{(x, q)\colon Rx \leq \lambda\Psi(q, x), Ax \leq b, x \geq 0, q \geq 0\},$$

such that $\mathcal{D}\colon \mathbb{R}^m_+ \times \mathbb{R}^m_+ \to 2^{\mathbb{R}^n_+ \times \mathbb{R}^m_+}$. Note that the correspondence is closed valued and the image of (λ, b) is a subset of $\mathscr{C} = \{(x, q)\colon Ax \leq b, q \leq M_2\}$, where existence of $M_2 < \infty$ follows from the assumption that G_i has a density with finite support. Since $\mathscr{C}$ is a compact set we have that $\mathcal{D}$ is also compact valued. Next we shall show that $\mathcal{D}$ is continuous, that is, both upper semicontinuous and lower semicontinuous.

For upper semicontinuity, consider any sequence $\{(\lambda_1, b_1), (\lambda_2, b_2), \ldots\}$ such that $(\lambda_n, b_n) \to (\lambda, b)$ as $n \to \infty$. Further consider any sequence $(x_n, q_n) \in \mathcal{D}(\lambda_n, b_n)$. Since (λ, b) is finite, there exists $M < \infty$ such that $(x_n, q_n) \leq M$. Thus, there exists a subsequence such that $(x_{n(m)}, q_{n(m)}) \to (x, y)$ as $m \to \infty$. Since G is atomless, we show that Ψ is continuous. Thus, we have $(x, q) \in \mathcal{D}(\lambda, b)$.

For lower semicontinuity, we will use the characterization given in Proposition 9.6, Sundaram (1996). Consider any closed set $F \in \mathbb{R}^n_+ \times \mathbb{R}^m_+$. Define

$$\underline{\lambda}_i := \sup\{(Rx)_i \psi_i^{-1}(q, x)\colon (q, x) \in F\},$$
$$\underline{b}_i := \sup\{(Ax)_i\colon (x, q) \in F\}.$$

It is easy to verify that the upper inverse of F under $\mathcal{D}$ is $\mathcal{D}^{-1}_+(F) = \{(\lambda, b)\colon \lambda_i \geq \underline{\lambda}_i$ for all $i = 1, \ldots, m$ and $b_k \geq \underline{b}_k$ for all $k = 1, \ldots, r\}$. Since $\mathcal{D}^{-1}_+(F)$ is closed, the correspondence $\mathcal{D}$ is lower semicontinuous. Thus, the correspondence $\mathcal{D}$ is compact valued and continuous. Further, continuity of Ψ in (43) implies that the objective function is continuous in (λ, x, q). Then, using the maximum theorem (cf. Sundaram 1996) we show that the point-to-set mapping Φ, defining the solution set of the optimization problem (43), is compact valued and upper semicontinuous. Further, Φ is a nonempty correspondence as $\{(0, 0)\} \in \mathcal{D}(\lambda, b)$ for all $(\lambda, b) \in \mathbb{R}^m_+ \times \mathbb{R}^r_+$.

The result follows by using a measurable selection theorem (see, e.g., Theorem 4, p. 342 of Cheney 2001). This completes the proof. □

References

Armony, M., C. Maglaras. 2004. On customer contact centers with a call-back option: Customer decisions, routing rules, and system design. *Oper. Res.* **52** 271–292.

Armony, M., N. Shimkin, W. Whitt. 2005. The impact of delay announcements in many-server queues with abandonment. Working paper, Stern School of Business, New York University, New York.

Bassamboo, A., A. Zeevi. 2006. On a data-driven method for staffing telephone call centers. *Oper. Res.* Forthcoming.

Bassamboo, A., J. M. Harrison, A. Zeevi. 2006a. Design and control of a large call center: Asymptotic analysis of an LP-based method. *Oper. Res.* **54** 419–435.

Bassamboo, A., J. M. Harrison, A. Zeevi. 2006b. Dynamic routing and admission control in high-volume service systems: Asymptotic analysis via multi-scale fluid-limits. *Queueing System Theory Appl.* **51** 249–285.

Bremaud, P. 1981. *Point Processes and Queues: Martingale Dynamics*. Springer Verlag, New York.

Cheney, W. 2001. *Analysis for Applied Mathematics*. Springer, New York.

de Véricourt, F., Y.-P. Zhou. 2005. Managing response time in a call-routing problem with service failure. *Oper. Res.* **53** 968–981.

Green, L., P. Kolesar. 1991. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Sci.* **37** 84–97.

Green, L. V., P. J. Kolesar, W. Whitt. 2007. Coping with time-varying demand when setting staffing requirements for a service system. *Production Oper. Management J.* **16**(1) 13–39.

Harrison, J. M. 2002. Stochastic networks and activity analysis. Y. Suhov, ed. *Analytic Methods in Applied Probability in Memory of Fridrih Karpelevich*. American Mathematical Society, Providence, RI, 53–76.

Harrison, J. M. 2003. A broader view of Brownian networks. *Annals Appl. Probability* **13** 1119–1150.

Harrison, J. M., A. Zeevi. 2005. A method for staffing large call centers based on stochastic fluid models. *Manufacturing Service Oper. Management* **7** 20–36.

Maglaras, C., A. Zeevi. 2005. Pricing and design of differentiated services: Approximate analysis and structural insights. *Oper. Res.* **53** 242–262.

Mandelbaum, A., W. A. Massey, M. Reiman. 1998. Strong approximations for Markovian service networks. *Queuing System Theory Appl.* **30** 149–201.

Massey, W. A., W. Whitt. 1998. Uniform acceleration expansions for Markov chains with time-varying rates. *Annals Appl. Probability* **8** 1130–1155.

Sundaram, R. K. 1996. *A First Course in Optimization Theory*. Cambridge University Press, Cambridge, UK.

Wein, L. M. 1991. Brownian networks with discretionary routing. *Oper. Res.* **39** 322–340.

Whitt, W. 1991. The pointwise stationary approximation for $M_t/M_t/s$ queues is asymptotically correct as the rates increase. *Management Sci.* **37** 307–314.

Whitt, W. 1999. Improving service by informing customers about anticipated delays. *Management Sci.* **45** 192–207.

Whitt, W. 2006. Fluid models for many-server queues with abandonments. *Oper. Res.* **54** 37–54.