WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

"Scheduling Networks of Queues:
Heavy Traffic Analysis
of a Multistation Network with Controllable Inputs"

Lawrence M. Wein

MIT Sloan School Working Paper #3046-89-MS

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

"Scheduling Networks of Queues:
Heavy Traffic Analysis
of a Multistation Network with Controllable Inputs"

Lawrence M. Wein

MIT Sloan School Working Paper #3046-89-MS

July 1989

Sloan School of Management,
Massachusetts Institute of Technology
Cambridge, MA 02139

SCHEDULING NETWORKS OF QUEUES: HEAVY TRAFFIC ANALYSIS OF A MULTISTATION NETWORK WITH CONTROLLABLE INPUTS

Lawrence M. Wein

Sloan School of Management, M.I.T.

Abstract

Motivated by a factory scheduling problem, we consider the problem of input control (subject to a specified input mix) and priority sequencing in a multistation, multiclass queueing network with general service time distributions and a general routing structure. The objective is to minimize the long-run expected average number of customers in the system subject to a constraint on the long-run expected average output rate. Under balanced heavy loading conditions, this scheduling problem can be approximated by a control problem involving Brownian motion. Linear programming is used to reduce the workload formulation of this control problem to a constrained singular control problem for a multidimensional Brownian motion. The finite difference approximation method is then used to find a linear programming solution to the latter problem. The solution is interpreted in terms of the original queueing system in order to obtain an effective scheduling policy. The priority sequencing policy is based on dynamic reduced costs from a linear program, and the workload regulating input policy releases a customer into the system whenever the workload process enters a particular region. An example is provided that illustrates the procedure and demonstrates its effectiveness.

Subject classification: Production/scheduling: priority sequencing in a stochastic job shop. Queues: Brownian models of network scheduling problems.

June 1989

SCHEDULING NETWORKS OF QUEUES: HEAVY TRAFFIC ANALYSIS OF A MULTISTATION NETWORK WITH CONTROLLABLE INPUTS

Lawrence M. Wein

Sloan School of Management, M.I.T.

Harrison [7] has introduced a Brownian system model that approximates a multiclass queueing network with dynamic scheduling capability. Under balanced heavy loading conditions, this model allows a queueing network scheduling problem to be approximated by a control problem involving Brownian motion. In Wein [20], a particular Brownian control problem was solved that approximates the problem of input control (subject to a specified product mix) and priority sequencing in a two-station multiclass queueing network with general service time distributions and a general routing structure. The objective was to minimize the long-run expected average number of customers in the system subject to a constraint on the long-run expected average output rate. The solution to the Brownian control problem was interpreted in terms of the original queueing system in Wein [21] in order to obtain an effective input control and priority sequencing policy.

In this paper we extend these results from the setting of a two-station network to a network with any finite number of stations. The two-station Brownian control problem was solved in [20] by (1) reformulating the problem in terms of workloads, (2) using linear programming to reduce the workload formulation to a constrained singular control problem for a one-dimensional Brownian motion, and (3) finding a closed-form solution to the constrained singular control problem. The resulting priority sequencing policy was based on dynamic reduced costs from the linear program, and the input policy depended on a two-dimensional workload process, which measured the total expected amount of work in the network for each of the two stations. This *workload regulating input policy* released a job into the network whenever the workload process entered a particular region in the

nonnegative orthant of R^2; this region was based on the solutions to both the linear program and the constrained singular control problem.

For the general multistation problem considered here, the Brownian control problem can again be reformulated in terms of workloads and linear programming can be employed to reduce the workload formulation to a constrained singular control problem. Therefore, the resulting sequencing policy is again based on dynamic reduced costs derived from the linear program. However, the constrained singular control problem now involves a multi-dimensional Brownian motion process, and the problem appears to be very difficult to solve in closed form. Instead, we employ the method of finite difference approximations (see Kushner [12] for a detailed development) to obtain a numerical solution to the constrained control problem. By discretizing both state and time, this technique allows us to approximate a controlled diffusion (and functionals of the controlled diffusion) by a controlled Markov chain (and functionals of the controlled Markov chain). In particular, if we ignore the constraints in our constrained singular control problem, then the problem can be approximated by a controlled Markov chain with a long-run average cost criterion. It is well-known (see, for example, Manne [15] and Derman [4]) that the latter problem can be formulated and solved as a linear program. Moreover, we show that under the finite difference approximation, the constraints in our singular control problem become linear constraints in the linear programming formulation of the Markov chain control problem. Therefore, these constraints can simply be added to the linear program for the Markov chain control problem, and an approximate solution to the constrained singular control problem can be found by solving a linear program. As in Wein [21], the proposed input policy is a workload regulating input policy, where the region of release depends on the solution to a linear program and the approximate solution to the constrained singular control problem.

In order to rigorously justify the finite difference approximation, one needs to prove (see Kushner [12]-[13]) that the optimally controlled Markov chain (suitably interpolated)

converges to the optimally controlled diffusion, and that the optimal cost of the controlled Markov chain converges to the optimal cost of the constrained singular control problem. Such a justification, which is typically based on weak convergence methods, is not attempted here. However, we do show that this finite difference approximation technique, when applied to the numerical example of a two-station queueing network scheduling problem considered in Wein [21], agrees with the closed form solution of this problem derived in Wein [20].

As in Wein [21], the customer release policy and the priority sequencing policy derived here work together in a coordinated way. The input policy reluctantly pulls work into the network, in that a customer is released whenever a server is threatened with idleness and there is not too much work already present in the network. The priority sequencing policy aggressively pushes this work to the various stations in order to avoid server idleness. As will be seen in Section 10, the proposed policies offer a significant improvement in performance over conventional customer release and priority sequencing policies.

This paper is organized as follows. In Section 1, the queueing network scheduling problem is described, and the workload formulation of the approximating Brownian control problem is given in Section 2. Linear programming is used in Section 3 to reduce the workload formulation to a constrained singular control problem, which is described in Section 4. The finite difference approximation, which allows the constrained singular control problem to be approximated by a constrained Markov chain control problem, is given in Section 5, and a linear programming solution to the latter problem is derived in Section 6. In Sections 7 and 8, respectively, the solution to the workload formulation is interpreted in terms of the original queueing system in order to obtain effective priority sequencing and input control policies, respectively, for the original queueing network scheduling problem. In Section 9, a heuristic extension to the input policy is described that allows for the decision of which class of customer to next release into the system, not just when to release the customer. An example is given in Section 10 that illustrates the entire procedure and

demonstrates its effectiveness, and concluding remarks are offered in Section 11.

1. The Queueing Network Scheduling Problem

The queueing network scheduling problem studied in this paper is motivated by a scheduling problem that is encountered in many factories; see Wein [21] for a description of the motivating factory scheduling problem. Consider a queueing network with I single-server stations and K customer classes. Class k customers require service at a particular station $s(k)$ and have service times that are independent and identically distributed random variables with mean m_k and variance s_k^2. A class k customer, upon completion of service at station $s(k)$, turns next into a class j customer with probability P_{kj} and exits the network with probability $1 - \sum_{j=1}^{K} P_{kj}$. The Markovian switching matrix P has spectral radius less than one, and so all customers eventually exit the network with probability one. Because the number of classes is allowed to be arbitrary, this routing structure is almost perfectly general.

There are input control and priority sequencing decisions in the scheduling problem. We assume there is an ample supply of customers who are waiting to gain entry into the network, and that each of these customers has an exogenously specified class designation. The class designations are such that q_k is the long-run proportion of class k customers released into the network. The *entering class mix* vector $q = (q_k)$ is such that $\sum_{k=1}^{K} q_k = 1$. We will assume that the class designations are deterministic; however, they could be Markovian without changing the nature of the analysis. The input decisions are to choose the non-decreasing process $N = \{N(t), t \geq 0\}$, where $N(t)$ is the cumulative number of customers released into the network in the time inteval $[0, t]$. Thus, the input decisions allow for full discretion over the timing of the release of customers into the system, but do not allow for the choice of which class of customer to release. In Section 9, we develop

a heuristic scheme that allows the controller to decide which class of customer to release into the system; this scheme guarantees that the actual mix that is released is sufficiently close to the desired mix q.

The sequencing decisions are to dynamically choose which class of customer to serve at each station in the network. Although preemptive resume scheduling is assumed, the assumptions regarding preemption do not have an impact on the proposed scheduling policy.

There is a per unit time holding cost c_k that is incurred when a class k customer is in the queueing network. Define the *throughput rate* of the queueing network to be the number of customer departures per unit of time. Then our queueing network scheduling problem is to find the input and sequencing policy that minimizes the long-run expected average holding costs subject to a constraint that the long-run expected average throughput rate is greater than or equal to the specified lower bound $\bar{\lambda}$. If $c_k = c$ for $k = 1, ..., K$, then the objective is to minimize the long-run expected average number of customers in the network. Since the throughput constraint will be tight in general, this latter objective is equivalent (by Little's formula) to minimizing the long-run expected average *cycle time* of customers, where a customer's cycle time is the length of time it spends in the network.

2. The Workload Formulation of the Brownian Control Problem

Harrison [7] has shown how the queueing network scheduling problem described in the last section is approximated by a control problem for a Brownian network. In Wein [20], it is shown how this Brownian control problem is reformulated in terms of workloads. Since the proposed scheduling policy depends only on the solution to the workload formulation, we will go directly to the workload formulation of the Brownian control problem that approximates the queueing network scheduling problem described in Section 1.

Let $\rho = (\rho_i)$ be the $I-$vector of *server utilizations*, or *traffic intensities*, for the I stations; later in this section, ρ will be explicitly defined in terms of the problem parameters. The Brownian approximation assumes the existence of a large integer n such that

$$\sqrt{n}(1 - \rho_i) \text{ is positive and of moderate size for } i = 1, ..., I. \tag{2.1}$$

Let $Q_k = \{Q_k(t), t \geq 0\}$ be the number of class k customers in the system at time t, for $k = 1, ..., K$, and let $I_i = \{I_i(t), t \geq 0\}$ be the cumulative amount of time that the server at station i is idle in the time interval $[0, t]$, for $i = 1, ..., I$.

With the parameter n fixed, define the scaled *queue length process* $Z_k = \{Z_k(t), t \geq 0\}$ by

$$Z_k(t) = \frac{Q_k(nt)}{\sqrt{n}}, \ t \geq 0 \text{ and } k = 1, ..., K, \tag{2.2}$$

and the scaled *cumulative idleness process* $U_i = \{U_i(t), t \geq 0\}$ by

$$U_i(t) = \frac{I_i(nt)}{\sqrt{n}}, \ t \geq 0 \text{ and } i = 1, ..., I. \tag{2.3}$$

Define the one-dimensional scaled centered *input* process θ by

$$\theta(t) = \frac{\bar{\lambda}nt - N(nt)}{\sqrt{n}}, \ t \geq 0, \tag{2.4}$$

where $N(t)$ is the cumulative number of customers released into the system up to time t and $\bar{\lambda}$ is the specified average throughput rate. The processes $Z = (Z_k), U = (U_i)$, and θ are the control processes in the workload formulation of the Brownian control problem.

Let the $K-$vector $\lambda = (\lambda_k)$ de defined by

$$\lambda = q\bar{\lambda}. \tag{2.5}$$

Since q is the entering class mix vector, it follows that λ_k represents the average number of class k customers that must depart from the system per unit of time in order to satisfy the throughput rate constraint. Define the $K \times K$ *input-output* matrix $R = (R_{kj})$ by

$$R_{kj} = m_j^{-1}(\delta_{jk} - P_{jk}), \tag{2.6}$$

where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ otherwise. The term R_{kj} represents the average rate at which class k customers are depleted when class j customers are being served. Since the routing matrix P is transient, the matrix R is nonsingular and there exists a unique nonnegative solution $\beta = (\beta_k)$ to the flow balance equations

$$\lambda = R\beta. \tag{2.7}$$

Define The $I \times K$ *resource consumption* matrix $A = (A_{ik})$ by

$$A_{ik} = \begin{cases} 1, & \text{if i = s(k);} \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

Then the server utilization vector ρ referred to in condition (2.1) is defined by

$$\rho = A\beta. \tag{2.9}$$

Now define the $I \times K$ *workload profile matrix* $M = (M_{ik})$ by

$$M = AR^{-1}. \tag{2.10}$$

The element M_{ik} represents the total expected remaining amount of work for a class k customer at station i until the customer exits the network. Let the $I-$dimensional *workload process* W be defined by

$$W(t) = MZ(t), \ t \geq 0, \tag{2.11}$$

so that $W_i(t)$ is interpreted as the total expected amount of scaled work anywhere in the network for station i at time t. Now define the $I-$vector $v = (v_i)$ by

$$v = Mq, \tag{2.12}$$

so that v_i is interpreted as the expected total amount of time over the long-run that server i spends on each customer. By Proposition 1 in Wein [21],

$$\rho_i = v_i\bar{\lambda} \text{ for } i = 1, ..., I. \tag{2.13}$$

Let the $I-$vector $\gamma = (\gamma_i)$ be defined by

$$\gamma_i = \sqrt{n}(1 - \rho_i) \text{ for } i = 1, ..., I. \tag{2.14}$$

Let $C(i)$ be the set of all customer classes k such that $s(k) = i$, and define the $K-$vector $\alpha = (\alpha_k)$ by

$$\alpha_k = \frac{\beta_k}{\rho_i} \text{ for all } k \in C(i). \tag{2.15}$$

Now define X to be a $K-$dimensional Brownian motion process with drift δ and covariance Σ, where

$$\delta = \sqrt{n}(\lambda - R\alpha) \tag{2.16}$$

and

$$\Sigma_{jl} = \sum_{k=1}^{K} [\alpha_k m_k^{-1} P_{kj}(\delta_{jl} - P_{kl}) + \alpha_k m_k^{-1} s_k^2 R_{jk} R_{lk}]. \tag{2.17}$$

Finally, let B be the $I-$dimensional Brownian motion process defined by

$$B(t) = MX(t), \ t \geq 0. \tag{2.18}$$

The process B has drift $M\delta$ and covariance $M\Sigma M^T$. It was shown in Wein [21] that the drift vector $M\delta = -\gamma$, where γ was defined in (2.14).

The Brownian control problem is obtained by letting the system parameter n defined in the balanced heavy loading condition (2.1) approach infinity. Exactly the same notation used for the scaled processes Z, U, and θ are used in defining the Brownian control problem in order to retain the queueing network interpretation of the Brownian model. Define the *workload formulation* of the Brownian control problem as choosing right continuous with left limit (RCLL) processes Z, U and θ (K-, I- and one-dimensional, respectively) so as to

$$\text{minimize} \quad \limsup_{T \to \infty} \frac{1}{T} E_x [\int_0^T \sum_{k=1}^{K} c_k Z_k(t) dt] \tag{2.19}$$

$$\text{subject to} \quad Z, U \text{ and } \theta \text{ are non} - \text{anticipating with respect to } X, \tag{2.20}$$

$$U \text{ is non} - \text{decreasing with } U(0) = 0, \tag{2.21}$$

$$Z(t) \geq 0 \text{ for all } t \geq 0, \tag{2.22}$$

$$\limsup_{T\to\infty} \frac{1}{T} E[U_i(T)] \leq \gamma_i \text{ for } i = 1, ..., I, \text{ and} \tag{2.23}$$

$$MZ(t) = B(t) + U(t) - v\theta(t) \text{ for all } t \geq 0. \tag{2.24}$$

Problem (2.19)-(2.24) is referred to as the workload formulation because the basic system state equation (2.24) is in terms of the $I-$dimensional workload process W defined in (2.11).

3. Solving For Z in Terms of U

In this section, we express the optimal process Z in the workload formulation in terms of the control process U. Suppose we are *given* a process U that satisfies constraints (2.20), (2.21), and (2.23). Then the optimal Z and θ processes are found by solving the following linear program at each time t:

$$\min_{Z(t),\theta(t)} \sum_{k=1}^{K} c_k Z_k(t) \tag{3.1}$$

$$\text{subject to } \sum_{k=1}^{K} M_{ik} Z_k(t) + v_i\theta(t) = B_i(t) + U_i(t) \text{ for } i = 1, ..., I, \tag{3.2}$$

$$Z_k(t) \geq 0, \text{ for } k = 1, ..., K. \tag{3.3}$$

At each time $t \geq 0$, this linear program may have a different set of right hand side values. The dual of this linear program will be easier to analyze, since it has a static constraint set. Let us define the dual variables $\pi(t) = (\pi_i(t))$ and state the *dual linear program* to be solved at time t:

$$\max_{\pi_1(t),...,\pi_I(t)} \sum_{i=1}^{I} [B_i(t) + U_i(t)]\pi_i(t) \tag{3.4}$$

$$\text{subject to} \quad \sum_{i=1}^{I} M_{ik}\pi_i(t) \leq c_k \quad \text{for } k = 1, ..., K, \tag{3.5}$$

$$\sum_{i=1}^{I} v_i \pi_i(t) = 0. \tag{3.6}$$

Before analyzing the dual LP (3.4)-(3.6), let us define the $(I-1)$-dimensional *workload imbalance process* $\hat{W} = (\hat{W}_i(t))$ by

$$\hat{W}_i(t) = \rho_I W_i(t) - \rho_i W_I(t), \ t \geq 0, \ \text{for } i = 1, ..., I-1. \tag{3.7}$$

By (2.11), (2.13), (2.24) and (3.7), the workload imbalance process can also be expressed as

$$\hat{W}_i(t) = \rho_I B_i(t) - \rho_i B_I(t) + \rho_I U_i(t) - \rho_i U_I(t), \quad t \geq 0, \ \text{for } i = 1, ..., I-1. \tag{3.8}$$

Thus, the workload imbalance process does not depend on the control process θ, and is expressed in terms of the Brownian motion process B and the control process U.

Returning to the dual LP (3.4)-(3.6), use (3.6) to eliminate $\pi_I(t)$ from the problem and use (2.13) to substitute the utilization levels ρ for the vector v to obtain the $(I-1)$-variable dual linear program

$$\max_{\pi_1(t),...,\pi_{I-1}(t)} \rho_I^{-1} \sum_{i=1}^{I-1} \hat{W}_i(t)\pi_i(t) \tag{3.9}$$

$$\text{subject to} \quad c_k^{-1} \sum_{i=1}^{I-1} (\rho_I M_{ik} - \rho_i M_{Ik})\pi_i(t) \leq \rho_I \quad \text{for } k = 1, ..., K. \tag{3.10}$$

Denote the solution to (3.9)-(3.10) by $(\pi_1^*(t), ..., \pi_{I-1}^*(t))$ and solve for $\pi_I^*(t)$ using equation (3.6). Let $\bar{c}_k(t)$ denote the *dynamic reduced cost* for the variable $Z_k(t)$ in the linear program (3.1)-(3.3). That is,

$$\bar{c}_k(t) = c_k - \sum_{i=1}^{I} \pi_i^*(t) M_{ik}. \tag{3.11}$$

We will return to the dynamic reduced costs in Section 4, where the costs become the basis for our proposed sequencing policy. By the analysis above, it follows that the linear program (3.1)-(3.3) can be expressed as

$$\min_{Z(t)} \sum_{k=1}^{K} c_k Z_k(t) \tag{3.12}$$

$$\text{subject to} \sum_{k=1}^{K} (\rho_I M_{ik} - \rho_i M_{Ik}) Z_k(t) = \hat{W}_i(t) \text{ for } i = 1, ..., I-1, \tag{3.13}$$

$$Z_k(t) \geq 0, \text{ for } k = 1, ..., K. \tag{3.14}$$

Denoting the solution to this linear program by $Z_k^*(t)$, we construct the optimal queue length process Z^* from $Z_k^*(t), k = 1, ..., K$, for all $t \geq 0$. Notice that this optimal process does not depend on the control process θ and depends on the control process U only through the workload imbalance process $\hat{W}$ defined in (3.7).

4. The Resulting Control Problem

In this section we substitute the optimal queue length process Z^* for Z into the workload formulation (2.19)-(2.24), and reduce this problem to one of finding the optimal $I-$dimensional cumulative idleness process U. By duality theory, it is known that the optimal value of the primal and dual objectives in problems (3.1)-(3.3) and (3.9)-(3.10) will be equal, or that

$$\sum_{k=1}^{K} c_k Z_k^*(t) = \rho_I^{-1} \sum_{i=1}^{I-1} \hat{W}_i(t) \pi_i^*(t). \tag{4.1}$$

Define the function $h : R^{I-1} \rightarrow R^1$ by

$$h(\hat{W}(t)) = \rho_I^{-1} \sum_{i=1}^{I-1} \hat{W}_i(t) \pi_i^*(t). \tag{4.2}$$

Then h is a piecewise-linear, continuous function of $\hat{W}(t)$, and h achieves a minimum of zero at the point $\hat{W}(t) = 0$.

Define the $(I-1)-$dimensional Brownian motion $\hat{B}$ by

$$\hat{B}_i(t) = \rho_I B_i(t) - \rho_i B_I(t),\ t \geq 0,\ \text{for } i = 1, ..., I-1. \tag{4.3}$$

Recalling that the two-dimensional Brownian motion B has drift $-\gamma$, it follows that $\hat{B}$ has drift $\mu = (\mu_i)$, where

$$\mu_i = \sqrt{n}(\rho_i - \rho_I),\ \text{for } i = 1, ..., I-1. \tag{4.4}$$

Notice that when the system is *perfectly balanced* (i.e., $\rho_i = \rho$ for $i = 1, ..., I$), then the drift $\mu = (0, ..., 0)$. Define the $(I-1) \times I$ matrix T by

$$T = \begin{pmatrix} \rho_I & 0 & 0 & . & . & 0 & -\rho_1 \\ 0 & \rho_I & 0 & . & . & . & -\rho_2 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & \rho_I & 0 & -\rho_{I-2} \\ 0 & 0 & 0 & . & 0 & \rho_I & -\rho_{I-1} \end{pmatrix}. \tag{4.5}$$

Then $\hat{B}$ has $(I-1) \times (I-1)$ covariance matrix $a = TM\Sigma M^T T^T$.

The resulting control problem is to find a non-anticipating (with respect to the $K-$dimensional Brownian motion X), non-decreasing, $I-$dimensional, RCLL process U to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E_x \left[\int_0^T h(\hat{W}(t))dt \right] \tag{4.6}$$

$$\text{subject to} \quad \limsup_{T\to\infty} \frac{1}{T} E_x[U_i(T)] \leq \gamma_i,\ \text{for } i = 1, ..., I, \tag{4.7}$$

$$\hat{W}_i(t) = \hat{B}_i(t) + \rho_I U_i(t) - \rho_i U_I(t),\ \text{for } i = 1, ..., I-1. \tag{4.8}$$

In problem (4.6)-(4.8), the controller observes the $(I-1)-$dimensional Brownian motion process $\hat{B}$ and exerts the non-anticipating $I-$dimensional control $U = (U_i)$. The constraint (4.7) determines an upper bound on the long-run expected average amount of control exerted. The resulting controlled process is the $(I-1)-$dimensional workload imbalance process $\hat{W}$, and since the optimal process Z^* derived in the previous section is

used, costs are incurred at a rate $h(\hat{W}(t))$. Notice that for $i = 1, ..., I-1$, the control U_i only affects the ith component of $\hat{W}$, whereas U_I affects all $I-1$ components of $\hat{W}$.

Problem (4.6)-(4.8) will be referred to as a *constrained singular control problem.* The name "singular" stems from the fact that the state of the controlled process can be instantaneously changed by the controller and, as a result, the optimal process U is continuous but singular (that is, the set of time points at which U increases has measure zero). Problem (4.6)-(4.8) can be viewed as a variant of the *finite-fuel* problem for Brownian motion, in which there is a constraint on the total amount of controlling effort that can be exerted (or fuel that can be consumed). In the traditional finite-fuel problem, the amount of control exerted is constrained to be finite over some finite or infinite time interval; see Beneš, Shepp, and Witsenhausen [1], Chow, Menaldi, and Robin [3], and Karatzas [10] for variants of this problem. In contrast, the constraints (4.7) are on the long-run expected average amount of control exerted.

If a solution U^* to the constrained control problem (4.6)-(4.8) can be found, then the optimal θ process is, via equation (3.2),

$$\theta^*(t) = v_1^{-1}[B_1(t) + U_1^*(t) - \sum_{k=1}^{K} M_{1k} Z_k^*(t)] \text{ for all } t \geq 0, \tag{4.9}$$

where $Z^*(t)$ is the solution to (3.12)-(3.14). The optimal process θ^* is never explicitly used to develop the proposed sequencing and input control policies.

We now show that the cost function h in the constrained singular control problem is convex.

Proposition 4.1. *The function $h(\hat{W}(t))$ defined in (4.2) is convex in $\hat{W}(t)$.*

Proof. By definition,

$$h(\hat{w}) = \max_{\pi_1, ..., \pi_{I-1}} \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i \pi_i \tag{4.10}$$

subject to

$$c_k^{-1} \sum_{i=1}^{I-1} (\rho_I M_{ik} - \rho_1 M_{Ik}) \pi_i \leq \rho_I \text{ for } k = 1, ..., K. \tag{4.11}$$

Let π^a be the solution to

$$\max_{\pi_1,...,\pi_{I-1}} \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^a \pi_i \tag{4.12}$$

subject to (4.11), and let π^b be the solution to

$$\max_{\pi_1,...,\pi_{I-1}} \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^b \pi_i \tag{4.13}$$

subject to (4.11). Thus, $h(\hat{w}^a) = \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^a \pi_i^a$ and $h(\hat{w}^b) = \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^b \pi_i^b$. For some $\lambda \in [0,1]$, let $\hat{w}^\lambda = (1-\lambda)\hat{w}^a + \lambda \hat{w}^b$, and suppose π^λ is the solution to

$$\max_{\pi_1,...,\pi_{I-1}} \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^\lambda \pi_i \tag{4.14}$$

subject to (4.11); thus, $h(\hat{w}^\lambda) = \rho_I^{-1} \sum_{i=1}^{I-1} [(1-\lambda)\hat{w}_i^a + \lambda \hat{w}_i^b] \pi_i^\lambda$. Therefore

$$(1-\lambda)h(\hat{w}^a) + \lambda h(\hat{w}^b) = (1-\lambda)\rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^a \pi_i^a + \lambda \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^b \pi_i^b \tag{4.15}$$

$$\geq (1-\lambda)\rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^a \pi_i^\lambda + \lambda \rho_I^{-1} \sum_{i=1}^{I-1} \hat{w}_i^b \pi_i^\lambda \tag{4.16}$$

$$= h(\hat{w}^\lambda), \tag{4.17}$$

where inequality (4.16) holds because π^λ satisfies constraint (4.11). ∎

The goal of the next two sections is to find a solution to the constrained singular control problem. To that end, we will make one more minor transformation of problem (4.6)-(4.8). The goal of this transformation is to eliminate the coefficients in front of the U_i terms in equation (4.8). In the perfectly balanced case where $\rho_i = \rho$ for $i = 1, ..., I$, the workload imbalance process $\hat{W}$ and the Brownian motion $\hat{B}$ could have been defined by $\hat{W}_i(t) = W_i(t) - \hat{W}_I(t) = \sum_{k=1}^{K} (M_{ik} - M_{Ik}) Z_k(t)$ and $\hat{B}_i(t) = B_i(t) - B_I(t)$ for $i = 1, ..., I-1$; then equation (4.8) could be expressed as $W_i(t) = \hat{B}_i(t) + U_i(t) - U_I(t)$.

When the system is not perfectly balanced, let $\bar{\rho}_i$ be defined by

$$\bar{\rho}_i = \prod_{\{j:j \neq i\}}^{I-1} \rho_j \text{ for } i = 1, ..., I-1. \tag{4.18}$$

and make the following definitions:

$$U_i^\circ(t) = \prod_{\{j:j\neq i\}}^{I} \rho_j U_i(t) \text{ and } \gamma_i^\circ = \prod_{\{j:j\neq i\}}^{I} \rho_j \gamma_i \text{ for } i = 1, ..., I; \tag{4.19}$$

$$\hat{W}_i^\circ(t) = \bar{\rho}_i \hat{W}_i(t) \text{ and } \hat{B}_i^\circ(t) = \bar{\rho}_i \hat{B}_i(t) \text{ for } i = 1, ..., I-1. \tag{4.20}$$

Also, define the function h° by $h^\circ(\hat{W}^\circ(t)) = h(\hat{W}(t))$. Notice that h° will be a piecewise linear, convex, and continuous function with a minimum of zero at zero. Then problem (4.6)-(4.8) can be expressed as choosing a non-decreasing, non-anticipative (with respect to X), $I-$dimensional, RCLL process U° to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E_x \left[\int_0^T h^\circ(\hat{W}^\circ(t)) dt \right] \tag{4.21}$$

$$\text{subject to} \quad \limsup_{T\to\infty} \frac{1}{T} E_x[U_i^\circ(T)] \leq \gamma_i^\circ, \text{ for } i = 1, ..., I, \tag{4.22}$$

$$\hat{W}_i^\circ(t) = \hat{B}_i^\circ(t) + U_i^\circ(t) - U_I^\circ(t), \text{ for } i = 1, ..., I-1. \tag{4.23}$$

For ease of notation, we shall drop all of the "$\circ$" superscripts and subsequently analyze the following problem: choose a non-decreasing, non-anticipating (with respect to X), $I-$dimensional, RCLL process U to

$$\text{minimize} \quad \limsup_{T\to\infty} \frac{1}{T} E_x \left[\int_0^T h(\hat{W}(t)) dt \right] \tag{4.24}$$

$$\text{subject to} \quad \limsup_{T\to\infty} \frac{1}{T} E_x[U_i(T)] \leq \gamma_i, \text{ for } i = 1, ..., I, \tag{4.25}$$

$$\hat{W}_i(t) = \hat{B}_i(t) + U_i(t) - U_I(t), \text{ for } i = 1, ..., I-1. \tag{4.26}$$

The drift and covariance of the Brownian motion $\hat{B}$ will be denoted by the $(I-1)-$dimensional vector $\mu = (\mu_i)$ and the $(I-1) \times (I-1)$ matrix $a = (a_{ij})$, respectively.

5. The Finite Difference Approximation Method

One possible approach to analyzing the constrained singular control problem is to form the Lagrangian relaxation of (4.24)-(4.26), where the constraints (4.25) are placed in the objective function and multiplied by the (unknown) Lagrange multipliers $l = (l_i)$:

$$\min_{U} \limsup_{T\to\infty} \frac{1}{T} E_x \left[\int_0^T h(\hat{W}(t))dt + \sum_{i=1}^{I} l_i U_i(T) \right] \tag{5.1}$$

$$\text{subject to } \hat{W}_i(t) = \hat{B}_i(t) + U_i(t) - U_I(t), \text{ for } i = 1, ..., I-1. \tag{5.2}$$

Under the case where the multipliers l are known, the Lagrangian problem has been analyzed by Taksar [19] for the case $I = 2$ and by Menaldi et al. [16] for general I. The optimality conditions for the Lagrangian problem are to find (V, g, U) such that

$$\text{Min } \{\Gamma V(x) + h(x) - g, l_1 + \frac{\partial V}{\partial x_1}, ..., l_{I-1} + \frac{\partial V}{\partial x_{I-1}}, l_I - \sum_{i=1}^{I-1} \frac{\partial V}{\partial x_i}\} = 0, \tag{5.3}$$

where

$$\Gamma = \frac{1}{2} \sum_{i=1}^{I-1} \sum_{j=1}^{I-1} a_{ij} \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{I-1} \mu_i \frac{\partial}{\partial x_i} \tag{5.4}$$

is the generator of $\hat{B}$. In (5.3), the *potential function* $V(x) : R^{I-1} \to R^1$ is the cost incurred under the optimal policy when the initial state of the controlled process $\hat{W}$ is x *minus* the cost incurred under the optimal policy when the initial state of $\hat{W}$ is zero. Also, the *gain* g appearing in (5.3) is the minimal average cost per unit time, independent of initial state.

An argument identical to Theorem 5.1 in Wein [20] shows that the constrained problem (4.24)-(4.26) can be solved by finding a solution (including the approximate multipliers l) to (5.3)-(5.4) that simultaneously satisfies constraints (4.25). For the special case $I = 2$, a closed form solution $(V, g, l_1, l_2, U_1, U_2)$ to (5.3)-(5.4) and (4.25) was found in Wein [20]. Although Menaldi et al. [16] have shown that (5.3)-(5.4) are sufficient conditions (along with some regularity conditions) for optimality of the Lagrangian problem (5.1)-(5.2) and have proven the existence of a solution (V, g, U) to (5.3)-(5.4), there appears to be little

hope of finding a closed form solution to (5.3)-(5.4). Therefore, finding a closed form solution (V, g, l, U) to (5.3)-(5.4) and (4.25) does not seem possible.

Thus, our goal is to obtain a *numerical* solution to the constrained singular control problem. However, finding a numerical solution (V, g, l, U) also appears to be an arduous task, since, even if a numerical solution (V, g, U) to (5.3)-(5.4) were obtained for some set of multipliers l (not an obvious task in itself), a search over the space of l would be needed to assure that U satisfied constraints (4.25). However, for the case $I = 2$, Wein [20] showed that the solution $(V, g, l_1, l_2, U_1, U_2)$ included any nonnegative pair (l_1, l_2) such that l_1 plus l_2 equaled a particular constant. If a solution (V, g, l, U) to (5.3)-(5.4), (4.25) included any nonnegative multipliers $(l_1, ..., l_I)$ such that $\sum_{i=1}^{I} l_i$ equaled a particular constant (I do not know if this is true), then a reduction in the space of l would be possible.

Considering these difficulties, a better approach to a numerical solution to this problem appears to be the *finite difference approximation* technique developed by Kushner [12]. The basic idea behind this method is to systematically approximate a controlled diffusion process by a controlled finite state Markov chain, and then to solve the resulting optimization problem for the controlled Markov chain. Weak convergence methods are then used to verify that the controlled diffusion process (and its corresponding optimal cost) can be approximated arbitrarily closely by the controlled Markov chain (and its optimal cost). In this paper, the finite difference method is described and two numerical examples are given, but no weak convergence proof is provided.

This method is particularly powerful for the control problem (4.24)-(4.26) because of the existence of constraints (4.25). Although very little is known about multi-dimensional stochastic control problems with side constraints using a dynamic programming formulation, the finite difference approximation can easily incorporate the side constraints (4.25), as will be shown in the next section.

We now begin to describe the finite difference method and, for ease of reference, we will retain most of the notation of Kushner [12]. When there are I stations in the network,

the state space of $\hat{W}$ in the control problem is R^{I-1}. In order to numerically solve the problem, we need to confine the process $\hat{W}$ to a bounded set, which will be denoted by G. The set G needs to be large enough so that $\hat{W}$ never reaches the boundary of G, which is denoted by ∂G. Define h, not to be confused with the cost function in (4.24), to be the *finite difference interval*, which dictates how finely the state space R^{I-1} is discretized. Let R_h^{I-1} be the finite difference grid on R^{I-1}; a point $x \in R_h^{I-1}$ if there exists integers $n_1, ..., n_{I-1}$ such that $x = \sum_{i=1}^{I-1} he_i n_i$, where e_i is the unit vector in the ith coordinate direction. The approximating controlled Markov chain, which we denote by $\{\xi_n^h, n \geq 0\}$, will have state space $G_h = R_h^{I-1} \cap G$. Before we define the transition probabilities for the controlled Markov chain, the controls, or *actions*, will be described.

Let $\mathcal{U}(x)$ be the action set for the controlled Markov chain $\{\xi_n^h, n \geq 0\}$ when it is in state $x \in G_h$. Recall that the actions in the Markov chain control problem correspond to the controls U in the singular control problem. Let $u_i = 1$ if the control corresponding to U_i is exerted, and let $u_i = 0$ if the control corresponding to U_i is not exerted, for $i = 1, ..., I$. Then action $u \in \mathcal{U}(x)$ is defined by $u = (u_1, ..., u_I)$, and the cardinality of the action set is 2^I for all $x \notin \partial G$. We will not concern ourselves with the action set and transition probabilities when $x \in \partial G$, since, in practice, the controlled Markov chain will never reach the boundary; one just enlarges the set G if the controlled Markov chain reaches ∂G.

Let $P^h(x, y; u)$ denote the transition probability from state x to state y when the action $u = (u_1, ..., u_I)$ is used in state $x \in G_h$. Define

$$Q_h = \sum_{i=1}^{I-1} a_{ii} - \frac{1}{2} \sum_{\{i,j:i \neq j\}}^{I-1} |a_{ij}| + h \sum_{i=1}^{I-1} |\mu_i| \tag{5.5}$$

and assume that

$$a_{ii} - \sum_{\{j:j \neq i\}}^{I-1} |a_{ij}| \geq 0 \text{ for } i = 1, ..., I-1. \tag{5.6}$$

As mentioned on page 92 of Kushner [12], if condition (5.6) does not hold, then a transformation to the principal vectors of a can be applied to assure (5.6) in a new coordinate

system. By definition of u, the action $u = (0, ..., 0)$ corresponds to exerting no control. In this case, define the transition probabilities $P^h(x, y; u)$ by

$$P^h(x, x \pm e_i h; u) = \frac{a_{ii} - \sum_{\{j:j\neq i\}}^{I-1} |a_{ij}| + 2h\mu_i^{\pm}}{2Q_h}, \tag{5.7}$$

$$P^h(x, x + e_i h + e_j h; u) = P(x, x - e_i h - e_j h; u) = \frac{a_{ij}^+}{2Q_h} \text{ for } i \neq j, \tag{5.8}$$

$$P^h(x, x - e_i h + e_j h) = P(x, x + e_i h - e_j h) = \frac{a_{ij}^-}{2Q_h} \text{ for } i \neq j, \tag{5.9}$$

and $P^h(x, y; u) = 0$ otherwise. Notice that the transition probabilities $P^h(x, y; u)$ are nonnegative and sum over y to one for each x.

When the action $u \neq (0, ..., 0)$, then the Markov chain transitions are deterministic, due to the exertion of controls. In this case,

$$P^h(x, x + \sum_{i=1}^{I-1} e_i u_i h - \sum_{i=1}^{I-1} e_i u_I h; u) = 1 \tag{5.10}$$

and $P^h(x, y; u) = 0$ otherwise.

Define the *interpolation interval* $\triangle t^h$ by

$$\triangle t^h = \frac{h^2}{Q_h}. \tag{5.11}$$

Equation (5.11) relates the size of the discretized time intervals to the size of the discretized space intervals and, together with the transition probabilities $P^h(x, y; u)$ in (5.7)-(5.10), assures that the first- and second-order moments of $\xi_{n+1}^h - \xi_h^n$, conditioned on ξ_h^n, are consistent with those of the controlled diffusion process $\hat{W}$.

Thus, the controlled process $\hat{W}$ described by equation (4.26) has been approximated by the controlled Markov chain $\{\xi_n^h, n \geq 0\}$ with transition probabilities given by (5.7)-(5.10). Since there are no costs on the controls exerted in problem (4.24)-(4.26), the cost incurred when action u is taken and the controlled Markov chain is in state x is simply given by $h(x)$, where the function h, which appears in objective (4.24), is defined in (4.2). Thus,

ignoring constraints (4.25) for the moment, our problem (4.24)-(4.26) is approximated by a problem of controlling a finite-state, finite action-set Markov chain with a long-run average cost criterion. In the next section, we show how to express the constraints (4.25) in terms of the approximating Markov chain control problem, and how to solve the resulting constrained Markov chain control problem.

6. A Linear Programming Solution

It is well known that a Markov chain control problem with a long-run average cost criterion can be formulated and solved as a linear program; readers are referred to Manne [15] and Derman [4] for early work on this subject. We begin this section by formulating the controlled Markov chain problem described in the last section, which ignored the constraints (4.25), as a linear program. Suppose that the cardinality of the set G_h is M, and let the states be indexed by $x = 1, ..., M$ and the actions indexed by $u = 1, ..., 2^I$. A stationary policy for the Markov chain control problem will be described by $\beta = (\beta_x(u), x = 1, ..., M; u = 1, ..., 2^I)$, where $\beta_x(u)$ equals the probability that action u is chosen when the controlled Markov chain is in state x. If $\beta_x(u)$ equals zero or one for all $x = 1, ..., M$ and $u = 1, ..., 2^I$, then β is referred to as a *pure* stationary policy; otherwise, β is referred to as a *randomized* stationary policy. Let $\pi = (\pi_{xu})$ denote the steady-state probability that the controlled Markov chain will be in state x and action u will be chosen if policy β is used; thus, π is policy-dependent and must satisfy

$$\pi_{xu} \geq 0 \text{ for } x = 1, ..., M; \ u = 1, ..., 2^I, \tag{6.1}$$

$$\sum_{x=1}^{M} \sum_{u=1}^{2^I} \pi_{xu} = 1, \text{ and} \tag{6.2}$$

$$\sum_{u=1}^{2^I} \pi_{yu} = \sum_{x=1}^{M} \sum_{u=1}^{2^I} \pi_{xu} P^h(x, y; u) \text{ for } y = 1, ..., M, \tag{6.3}$$

where the transition probabilities $P^h(x, y; u)$ were given in equations (5.7)-(5.10). Since

the expected average cost under policy β is $\sum_{x=1}^{M}\sum_{u=1}^{2^I}\pi_{xu}h(x)$, the Markov chain control problem is to find $\pi = (\pi_{xu})$ to

$$\min \sum_{x=1}^{M}\sum_{u=1}^{2^I}\pi_{xu}h(x) \tag{6.4}$$

subject to constraints (6.1)-(6.3). If $\pi^* = (\pi^*_{xu})$ solves the linear program (6.1)-(6.4), then the optimal policy β^* is

$$\beta^*_x(u) = \frac{\pi^*_{xu}}{\sum_{u=1}^{2^I}\pi^*_{xu}}. \tag{6.5}$$

Thus, the linear program (6.1)-(6.4) solves the problem (4.24)-(4.26) that ignores constraints (4.25). We now show that under the finite difference approximation, the I constraints (4.25) can be expressed as I linear constraints on $\pi = (\pi_{xu})$. From equation (5.10), if the control u in the controlled Markov chain $\{\xi^h_n, n \geq 0\}$ is such that $\mu_i = 1$, then the corresponding cumulative amount of control exerted by U_i in problem (4.24)-(4.26) is increased by the finite difference interval size h. Equation (5.10) also implies that the steady-state probability that $u_i = 1$ is $\sum_{x=1}^{M}\sum_{\{u:u_i=1\}}^{2^I}\pi_{xu}$. Thus, by the finite difference approximation, the term $\limsup_{T\to\infty} T^{-1}E_x[U_i(T)]$ in (4.25) is approximated by

$$\lim_{n\to\infty}\frac{nh\sum_{x=1}^{M}\sum_{\{u:u_i=1\}}^{2^I}\pi_{xu}}{n\Delta t} \text{ for } i = 1,...,I, \tag{6.6}$$

which equals

$$\frac{Q_h\sum_{x=1}^{M}\sum_{\{u:u_i=1\}}^{2^I}\pi_{xu}}{h} \tag{6.7}$$

by the critical relationship (5.11) relating the time and space intervals. Therefore the constraints (4.25) are approximated by

$$\sum_{x=1}^{M}\sum_{\{u:u_i=1\}}^{2^I}\pi_{xu} \leq \frac{h\gamma_i}{Q_h} \text{ for } i = 1,...,I. \tag{6.8}$$

Thus, a solution to the constrained Markov chain control problem and an approximate solution to the constrained singular control problem (4.24)-(4.26) can be obtained by solving the linear program (6.1)-(6.4), (6.8). The solution to this linear program will yield an

optimal policy β for which the controls for up to I states are randomized. Since the cost function h is convex and achieves a minimum at zero, the optimal policy will be characterized by a *bounded region* containing the origin. This region is such that $u = (0, ..., 0)$ inside this region and $u \neq (0, ..., 0)$ outside of this region. The boundary of the region acts as a reflecting barrier that keeps the controlled process within the region containing the origin. The bounded region will play a key role in defining the customer release policy in Section 8.

The linear program (6.1)-(6.4), (6.8) suffers from the curse of dimensionality that is so often encountered in control problems. For example, if the grid G_h is such that each coordinate dimension is discretized into $L-1$ segments, then the cardinality of G_h is L^{I-1} and the linear program has $2^I L^{I-1}$ variables and $L^{I-1}+I+1$ constraints, not including the nonnegativity constraints (6.1). The size of this problem can be reduced in several ways. Since the holding costs are convex and have a minimum at zero, any control $u \neq (0, ..., 0)$ that pushes the controlled process further from the origin will clearly be suboptimal and need not be explicitly considered in the linear program. Also, the linear program can be solved several times (using smaller values of the finite difference interval h for later runs), and portions of the state space that the controlled process never reaches can be eliminated from the subsequent linear programs, as long as the controlled process never reaches the boundary of G_h. Moreover, the finite difference interval h can be state-dependent, so that a finer grid can be used in the sensitive portion of the grid G_h (where the optimal control is uncertain) and a courser grid can be used in the insensitive portion (where the optimal control is known). Finally, some decomposition approaches used in linear programming may be used to exploit the structure of the constraint set (in particular, the structure in (5.7)-(5.10) that appears in (6.3)); readers are referred to Kushner and Chen [14] for work on this topic.

We finish this section with a numerical example that illustrates the accuracy of the finite difference approximation for solving the constrained singular control problem (4.24)-

(4.26). The example is taken from Section 7 of Wein [21] and is derived from a particular two-station queueing network scheduling problem; the scheduling problem is identical to the problem considered in this paper with $I = 2$ stations. The cost function $h(x)$ in (4.24) is given by

$$h(x) = \begin{cases} -h_1 x, & \text{if } x < 0, \\ h_2 x, & \text{if } x \geq 0, \end{cases} \tag{6.9}$$

where $h_1 = .101$ and $h_2 = .3703$. The one-dimensional Brownian motion $\hat{B}$ has drift $\mu = 0$ and variance $a = 10.93$. The righthand side values in (4.25) are $\gamma_1 = \gamma_2 = .9$. The closed form solution to this problem (see Wein [20]) is characterized by the *interval endpoints* l and r, where

$$l = -\frac{h_2}{h_1 + h_2} \frac{a}{2\gamma_1} = -4.771 \tag{6.10}$$

and

$$r = \frac{h_1}{h_1 + h_2} \frac{a}{2\gamma_1} = 1.301. \tag{6.11}$$

The controlled process $\hat{W}$ behaves as a one-dimensional reflected, or regulated, Brownian motion (see Harrison [6] for a detailed development) on the interval $[-4.771, 1.301]$ and the optimal control processes U_1^* and U_2^* are the local times at the points -4.771 and 1.301, respectively. The optimal objective function value is given by (see Wein [20])

$$\frac{a h_1 h_2}{4\gamma_1 (h_1 + h_2)} = .2409. \tag{6.12}$$

To formulate the linear program (6.1)-(6.4), (6.8), a finite difference interval of size $h = 0.05$ was used and the bounded set G was taken to be the interval $[-5.0, 5.0]$. Thus the state space is $\{-5.0, -4.95, -4.90, ..., 4.95, 5.0\}$, and the states are indexed by $x = 1, ..., 201$. The action set $\mathcal{U}(x)$ consists of the four actions $(0,0), (1,0), (0,1)$, and $(1,1)$, which correspond, respectively, to exerting no control, exerting only U_1, exerting only U_2, and exerting both U_1 and U_2. The non-zero transition probabilities are

$$P(x, x+1; 0, 0) = P(x, x-1; 0, 0) = \frac{1}{2}, \tag{6.13}$$

$$P(x, x+1; 1, 0) = P(x, x-1; 0, 1) = 1, \text{ and} \tag{6.14}$$

$$P(x, x; 1, 1) = 1. \tag{6.15}$$

Of course, control $u = (1,1)$ will never be used and control $u = (1,0)(u = (0,1)$, respectively) will never be used when the controlled process is positive (negative, respectively). Since $Q_h = a$ by (5.5), constraints (6.8) are given by

$$\sum_{x=1}^{201} \sum_{\{u:u_i=1\}}^{4} \pi_{xu} \leq \frac{(.05)(.9)}{10.93} = .00412 \text{ for } i = 1,2. \tag{6.16}$$

Rather than indexing the states by $x = 1, ..., 201$, let us denote the state of the controlled Markov chain by $y \in [-5.0, -4.95, ..., 4.95, 5.0]$. The solution to the linear program yields, via equation (6.5),

$$\beta_y^*(0,0) = 1 \text{ for } y \in [-4.7, 1.25], \tag{6.17}$$

$$\beta_{-4.8}^*(1,0) = 1, \tag{6.18}$$

$$\beta_{4.75}^*(1,0) = .327, \tag{6.19}$$

$$\beta_{4.75}^*(0,0) = .673, \text{and} \tag{6.20}$$

$$\beta_{1.30}^*(0,1) = 1. \tag{6.21}$$

The optimal objective function value for the linear program was .2411, which is very close to the derived value of .2409 appearing in equation (6.12). If we interpolate the policy β at $y = -4.75$, the approximate solution to the constrained singular control problem is characterized by the interval $[-4.784, 1.300]$, where the controlled process $\hat{W}$ behaves as a one-dimensional reflected Brownian motion on this interval and U_1 and U_2 are the local times of $\hat{W}$ at the two respective endpoints. The interval derived from the finite difference approximation $[-4.784, 1.300]$ is very close to the exact interval $[-4.771, 1.301]$ at the rather modest finite difference interval size of $h = .05$. This particular example was also solved at the finer interval size of $h = .01$. The optimal objective function value for the linear program was again .2411, and the interpolated interval was [-4.777,1.300]. Thus, the finite difference approximation method is very accurate, at least for the simple case where a known solution exists.

7. The Sequencing Policy

In this section we will interpret the solution to the workload formulation in order to obtain a priority sequencing policy to the original queueing network scheduling problem. As in Harrison [7], Harrison and Wein [8], and Wein [21], the policy will be interpreted in terms of the optimal control process Z^*, where $Z_k^*(t)$ is interpreted as the (scaled) number of class k customers in the system at time t; readers are referred to these previous works for a more detailed discussion on the interpretation of the solutions to Brownian control problems. In particular, the sequencing policy is based on the dynamic reduced costs $\bar{c}_k(t)$ computed in (3.11). The reduced cost for a class k customer at time t is the increase in the optimal objective function value of the linear program (3.1)-(3.3) per unit increase in the righthand side of the nonnegativity constraint $Z_k(t) \geq 0$. This value can be interpreted as the extra cost incurred if one were forced to hold a class k customer in queue at time t. Thus, the higher the value of $\bar{c}_k(t)$, the more expensive it is to hold a class k customer in the system at time t.

However, each customer class requires a different amount of expected processing time before exiting. Therefore, it is natural to consider the amount of effort needed to completely process a customer, in addition to the cost incurred in holding the customer. As pointed out by Yang [24], the ratio

$$\frac{\bar{c}_k(t)}{\sum_{i=1}^{I} M_{ik}} \tag{7.1}$$

measures how costly a class k customer is at time t per unit of remaining processing time. Our proposed policy is to give priority at each station to the customer class with the largest value of this dynamic index. Yang [24] has shown that this policy, when applied to the Brownian analysis of a single-server multiclass queue with per unit time holding cost c_k for class k customers (see also Section 6 of Harrison [7]), reduces to the well-known $c\mu$ rule (see, for example, Klimov [11]).

Complementary slackness implies that the reduced cost $\bar{c}_k(t)$ equals zero when $Z_k^*(t) > 0$ in (3.1)-(3.3). Thus, the policy proposed in (7.1) will serve a customer from classes with $Z_k^*(t) > 0$ only when there are no customers present from classes with $Z_k^*(t) = 0$. Our policy is therefore consistent with observations from existing heavy traffic limit theorems (see, for example, Whitt [22], Harrison [5], Reiman [18], Johnson [9], Peterson [17], and Chen and Mandelbaum [2]) that the normalized queue length process of the lowest priority customers is positive and the normalized queue length processes of the higher priority customers vanish; of course, these results assume static (i.e., independent of the state of the queueing system) priority rankings, whereas we are proposing dynamic (i.e, state-dependent) priority rankings.

Notice that it is possible for several different customer classes with $Z_k^*(t) > 0$ (and therefore with $\bar{c}_k(t) = 0$) to be served at a common station. At times when only these customers are present at a particular station, a tie-breaking rule is needed to decide which of these classes to serve next. We employ the tie-breaking rule proposed in Yang [24], which attempts to reduce the total expected holding cost $\sum_{k=1}^{K} c_k Z_k(t)$ along a steepest descent direction. Readers are referred to that paper for a derivation of this heuristic tie-breaking rule; only a description of the rule will be presented here.

Define the $(I-1) \times K$ *workload imbalance profile* matrix $\hat{M} = (\hat{M}_{ik})$ by

$$\hat{M}_{ik} = \rho_I M_{ik} - \rho_i M_{Ik} \quad \text{for } i = 1, ..., I-1; k = 1, ..., K. \tag{7.2}$$

Then constraint (3.13) can be expressed as $\sum_{k=1}^{K} \hat{M}_{ik} Z_k(t) = \hat{W}_i(t)$ for $i = 1, ..., I-1$. Let $\hat{M}_B(t)$ be the submatrix of the matrix $\hat{M}$ consisting of the columns that are in the optimal basis of the linear program (3.12)-(3.14) at time t. It follows that $\hat{M}_B(t)$ is an $(I-1) \times (I-1)$ invertible matrix for all t. Let the $(I-1)$−dimensional vector $\Delta Z(t)$ be defined by

$$\Delta Z(t) = M_B^{-1}(t) \pi^*(t), \tag{7.3}$$

where $\pi^*(t) = (\pi_1^*(t), ..., \pi_{I-1}^*(t))$ is the optimal solution to the dual LP (3.9)-(3.10). If

class k is in the optimal basis at time t, then we denote its component of $\Delta Z(t)$ in (7.3) by $\Delta Z_k(t)$ in the obvious way. The dynamic tie-breaking rule is to give higher priority at time t to the classes with the higher values of $\Delta Z_k(t)$.

To summarize the priority sequencing policy, we compute a dynamic reduced cost $\bar{c}_k(t)$ for each class k at each time t using (3.11). If class k is in the optimal basis at time t, then $\bar{c}_k(t) = 0$ and a secondary index $\Delta Z_k(t)$ is computed according to equation (7.3). Recall that $C(i)$ is the set of customer classes that can be served at station i. At time t, server i gives priority to the customer class $k \in C(i)$ with the largest value of $\bar{c}_k(t)$. If $\bar{c}_k(t) = 0$ for all customers present at station i at time t, then server i gives priority to the customer class with the largest value of $\Delta Z_k(t)$.

8. The Input Policy

In this section the solution (Z^*, U^*, θ^*) to the workload formulation (2.19)-(2.24) is interpreted in order to obtain an input, or customer release, policy for the original queueing system. The input policy will be interpreted in terms of all three controls, in contrast to the sequencing policy, which was interpreted solely in terms of the optimal queue length process Z^*.

We begin by interpreting the Z^* control. Notice that only $I - 1$ constraints in the dual LP (3.9)-(3.10) will be tight at any time t, and thus, by complementary slackness, only $I - 1$ components of the optimal control process Z^* can be positive at any time t. Therefore, since $W(t) = MZ^*(t)$ for all $t \geq 0$, the $I-$dimensional workload process W stays on the boundary of a polyhedral cone in the nonnegative orthant of R^I. This cone, which is not necessarily convex, is generated by a number of extreme rays emanating from the origin, where each ray corresponds to a customer class that is in the optimal basis of the LP (3.12)-(3.14) for some value of $\hat{W}(t)$. Thus the number of different extreme rays,

and hence the number of different $(I-1)$−dimensional faces of the cone, equals the number of extreme points of the dual constraint set (3.10). See Figure 1 for an example with three stations (that is, $I = 3$) and six extreme rays.

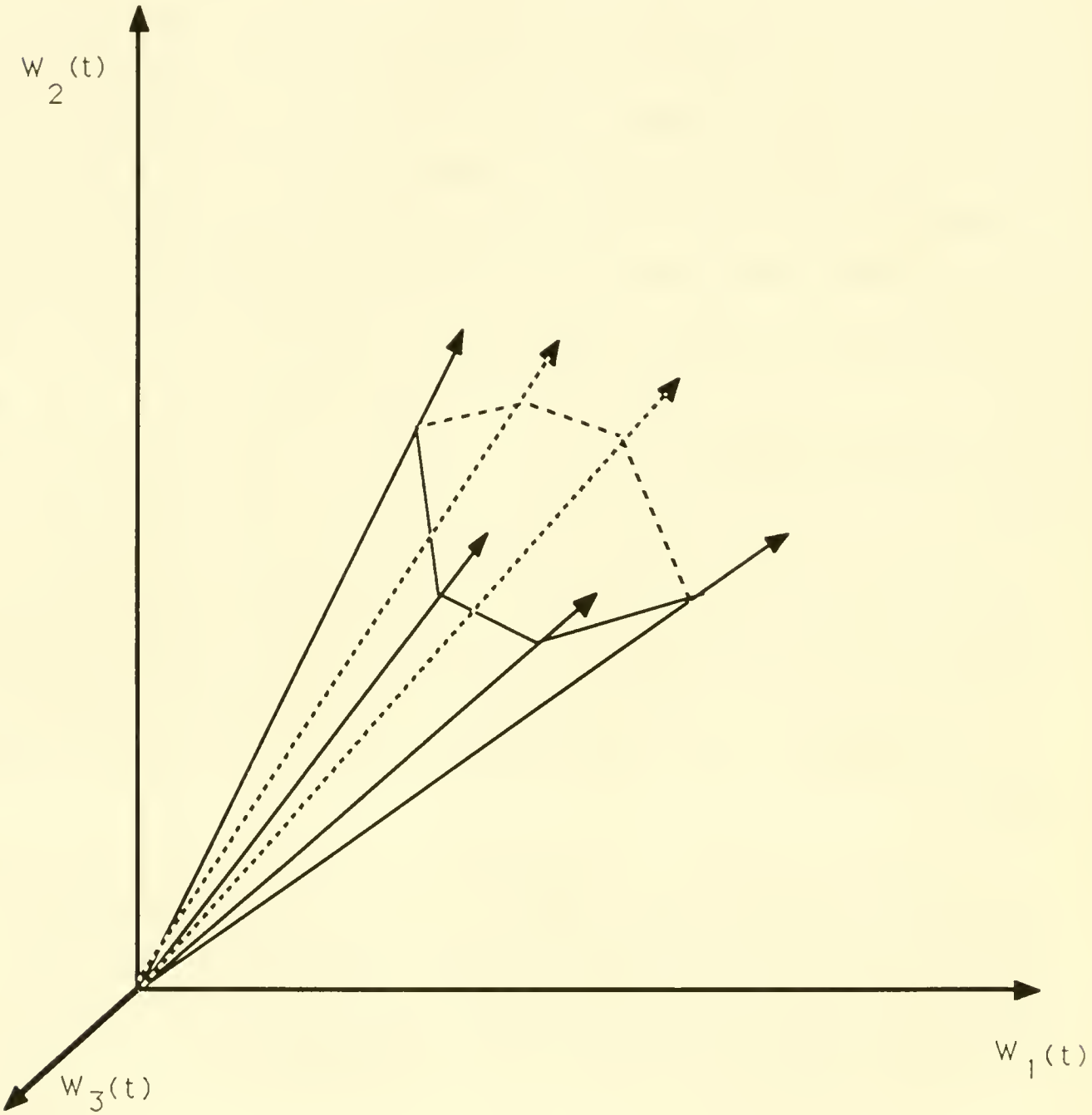

FIGURE 1. $W(t)$ Stays on the Cone Boundary.

The workload process W can also be expressed in terms of the other two controls, U^* and θ^*. By equations (2.11) and (2.24), the controller in the workload formulation observes an $I-$dimensional Brownian motion process B, exerts the optimal controls U^*, which is the $I-$dimensional scaled cumulative idleness process, and θ^*, which is the one-dimensional scaled centered input process, and obtains the controlled process W, which is the scaled workload process, via the equations

$$W_i(t) = B_i(t) + U_i^*(t) - v_i\theta^*(t), \text{ for } i = 1, ..., I \text{ and } t \geq 0. \tag{8.1}$$

Recall that the solution U^* to the constrained singular control problem is characterized by a bounded region in R^{I-1} containing the origin. The boundary of the region acts as a reflecting barrier that keeps the $(I-1)-$dimensional workload imbalance process $\hat{W}$ within this region containing the origin. Moreover, the control process U^* is exerted only when the workload imbalance process $\hat{W}$ reaches the reflecting barrier. Exerting the control U_i^* is interpreted as incurring server idleness at station i, for $i = 1, ..., I$.

The restriction of the $(I-1)-$dimensional workload imbalance process $\hat{W}$ to a bounded region containing the origin implies the restriction of the $I-$dimensional workload process W to a finite portion of the cone boundary. Thus the reflecting boundary in R^{I-1}, which was derived in the solution to the constrained singular control problem, essentially truncates the polyhedral cone in R^I. The intersection in R^I of the boundary of the original polyhedral cone and the reflecting barrier will be referred to as the *upper edge* of the truncated cone. See Figures 2 and 3 for typical cases when $I = 2$ and 3, respectively. In Figure 2, the reflecting boundary in R^1 is the interval $[a, b]$, and the upper edge of the truncated cone consists of two points. The upper edge of the truncated cone is a curve in R^3 in Figure 3 and, in general, is of dimension $I - 2$.

We can now make the following three observations about the optimal solution to the workload formulation: (1) the $I-$dimensional workload process W stays on the truncated cone boundary, (2) the control process U^* is exerted only when the workload process W

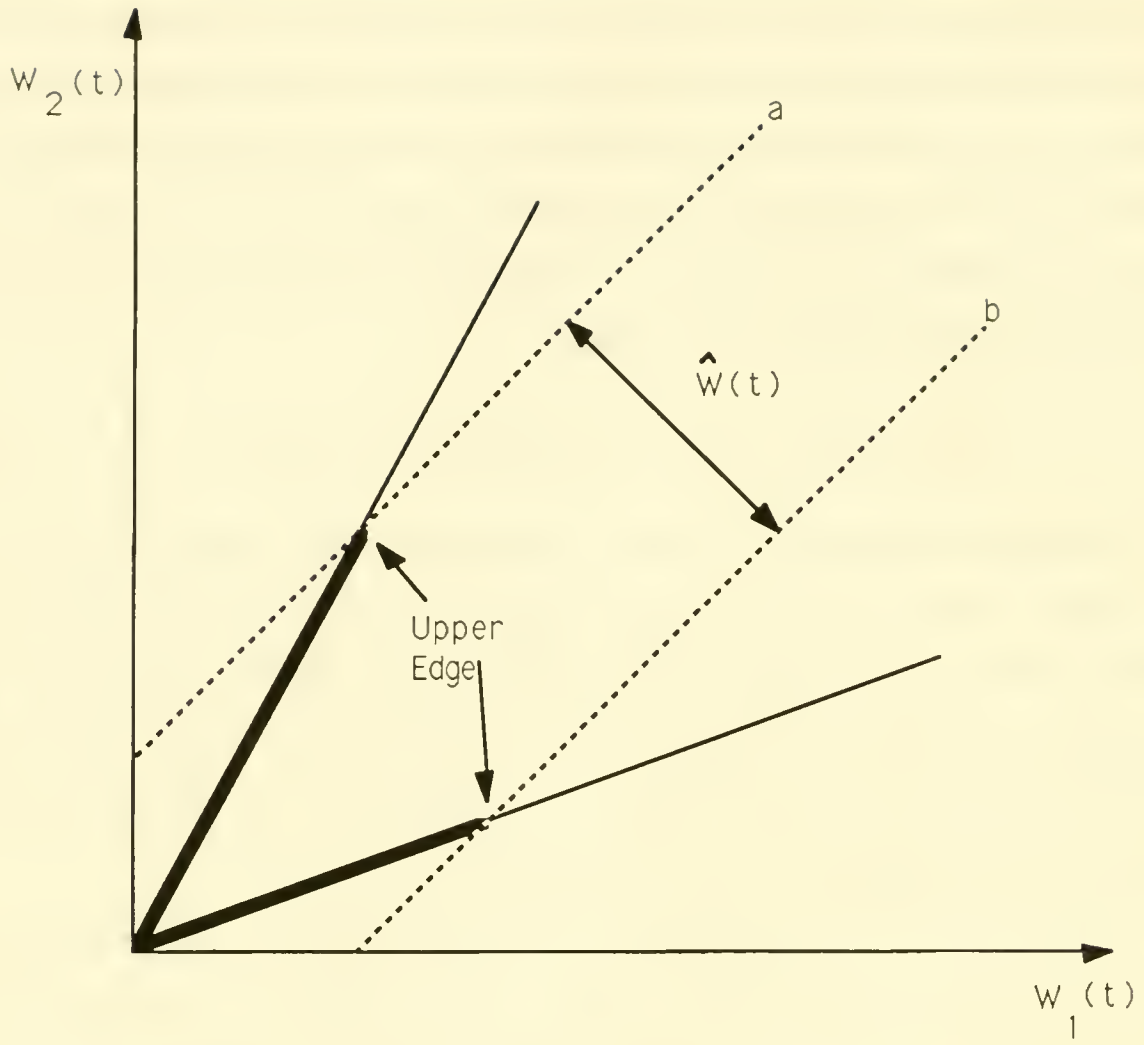

FIGURE 2. The Truncated Cone in a Two-Station Example.

reaches the upper edge of the truncated cone, and (3) when the workload process is not at the upper edge of the truncated cone, then only the input process θ^* is used to keep the workload process on the truncated cone boundary.

The goal of this section is to develop a customer release policy for the original queueing system that operationalizes these three observations. All these observations are developed under the idealized assumptions of the balanced heavy loading condition (2.1). Notice that in the actual queueing system, the workload process may not reside exactly on the truncated cone boundary. The actual workload process may reside in the interior of the

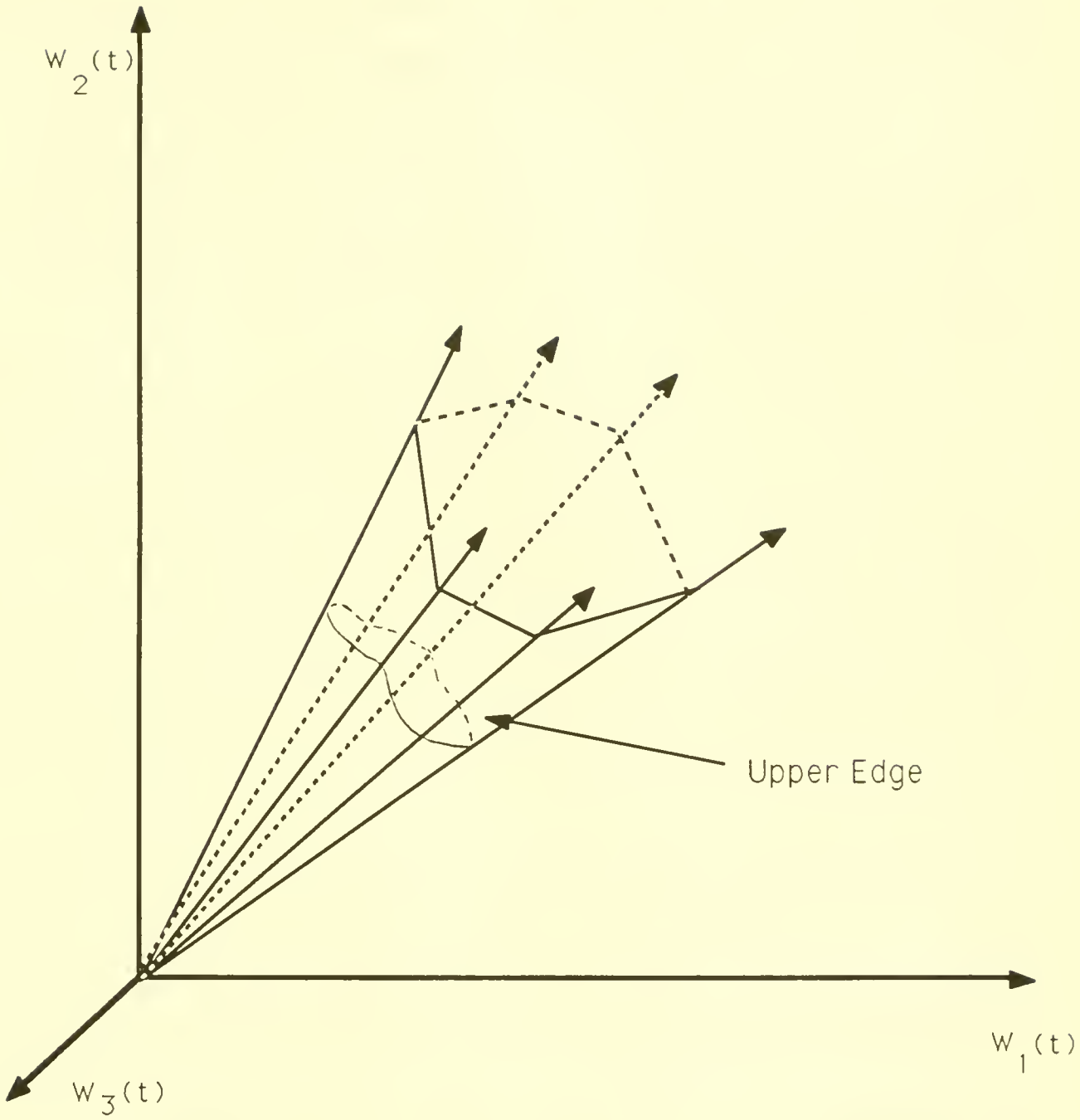

FIGURE 3. The Truncated Cone in a Three-Station Example.

truncated cone or, since the extremal rays of $\{W|W = MZ, Z \geq 0\}$ may not coincide with the extremal rays generating the truncated cone, may reside outside the truncated cone.

We now attempt to operationalize the control process θ^*. In order to see how the scaled

centered input process θ^* is manipulated, recall that by equation (2.13) and the balanced heavy loading condition (2.1), v_i is approximately equal to v_j for $i, j = 1, ..., I$, and so θ^* can move along a direction that is close to the $(1, 1, ..., 1)$ direction in R^I. By definition (2.4), when θ^* is increased and moves in the direction that is close to $(1, 1, ..., 1)$, input is being increased relative to the nominal input rate $\bar{\lambda}$. Similarly, when θ^* is decreased and moves in the direction that is close to $(-1, -1, ..., -1)$, input is being withheld relative to the nominal input rate.

When the workload process W is in the cone interior and not near the upper edge, then θ^* is decreased in order to keep W on the truncated cone boundary. Similarly, when the workload process W is outside the cone and not near the upper edge, then θ^* is increased in order to keep W on the truncated cone boundary. As in Wein [21], let us interpret the action "increase θ^*" to simply mean "release a customer into the system" and the action "decrease θ^*" to mean "cease input". The naive policy that emerges from this interpretation and from observations (1)-(3) above is to only release a customer into the system at times t when the workload process $W(t)$ is on the outside of the truncated cone.

A precise definition of what is meant by "outside the truncated cone" will be given shortly, but first the insights behind observations (1)-(3) will be summarized. When the workload process leaves the interior of the truncated cone, then the workload process becomes too imbalanced (recall the importance of the workload imbalance process $\hat{W}$) and at least one server becomes threatened with idleness. In this case, a customer is released into the network to avoid server idleness. However, when the workload process reaches the upper edge of the truncated cone, then there is already ample work in the system and the controller refuses to release any more customers into the system and is willing to incur server idleness.

We now return to the issue of defining the term "outside the truncated cone". Let R_B denote the bounded region in R^{I-1} that characterizes the solution to the constrained

singular control problem in Section 6. Recall that the number of faces on the boundary of the polyhedral cone in R^I is equal to the number of extreme points in the dual constraint set (3.10). Let us denote this number by F, and index the faces, and hence the dual extreme points, by $f = 1, ..., F$. For each face f, let us define $R_f \subset R^{I-1}$ to be the region in the workload imbalance space where extreme point f is the optimal solution to the dual LP (3.9)-(3.10). For dual extreme point f, let the customer classes in the optimal basis of the primal LP be indexed by $k_1, ..., k_{I-1}$. Then the hyperplane generated by cone face f is given by

$$\sum_{i=1}^{I} a_i W_i(t) = 0, \tag{8.2}$$

where

$$(a_1 W_1(t), ..., a_I W_I(t)) = \det \begin{vmatrix} W_1(t) & \dots & W_I(t) \\ M_{1k_1} & \dots & M_{Ik_1} \\ \vdots & \dots & \vdots \\ M_{1k_{I-1}} & \dots & M_{Ik_{I-1}} \end{vmatrix}. \tag{8.3}$$

Let r_f denote the halfspace in R^I (lying outside of the cone) generated by the hyperplane (8.2)-(8.3).

The *workload regulating* input policy will be to release a customer into the network whenever the $I-$dimensional workload process $W(t)$ enters the release region $\cup_{1 \le f \le F} \bar{R}_f$, where

$$\bar{R}_f = \{W(t) | (\hat{W}(t) \in R_f \cap R_B) \cap (W(t) \in r_f \cap R_I^+)\}, \tag{8.4}$$

and $\hat{W}(t)$ is defined in terms of $W(t)$ in (3.7); this region defines the vague term "outside the truncated cone".

In order to motivate policy (8.4), it is easiest to consider a two-station example, as in Figure 4, where the upper edge is given by two points; let us denote the lower point by (x_1, y_1) and the upper point by (x_2, y_2), where $x_1 > y_1$ and $x_2 < y_2$. For the sake of concreteness and without loss of generality, suppose the queueing network generating Figure 4 was a balanced system, so that $v_1 = v_2$ and θ^* moves in the $(1, 1)$ direction or the $(-1, -1)$ direction; then the reflecting boundary $[a, b]$ introduced in Figure 2 is

$[x_2 - y_2, x_1 - y_1]$. Notice that as long as the workload process W is in the darkly shaded region in Figure 4 (the intersection of the darkly shaded regions and the gray regions are along the 45 degree lines), then θ^* alone can be increased to move W to the truncated cone boundary. When $W(t) = (x_1, y)$, where $y < y_1$, then U_2^* alone is used to move the workload process to the truncated cone boundary, in this case to the point (x_1, y_1). In the lower (respectively, upper) gray region of Figure 2, some combination of θ^* and U_2^* (respectively, U_1^*) is used to move W to the truncated cone boundary. Therefore, the release region for this example should be at least the darkly shaded regions and at most the union of the darkly shaded regions and the gray regions. Although either extreme (or some compromise between the two extremes) would probably lead to an effective customer release policy, we have proposed in (8.4) the minimum release region, which in this example corresponds to the darkly shaded regions. This choice does not maintain consistency with previous work (the policy proposed in Wein [21] was the maximal release region, which corresponds to the union of the darkly shaded regions and the gray regions in Figure 4); however, the minimal release region suggested here is more easily generalizable to higher dimensions.

For the example in Figure 4, the reflecting boundary R_B is given by $\hat{W}(t) \in [a, b]$, or $y_2 - x_2 \leq W_1(t) - W_2(t) \leq y_1 - x_1$. If the upper face (respectively, lower face) is denoted by face 1 (respectively, face 2), then (see Wein [20]) R_1 is $\hat{W}(t) < 0$, or $\frac{W_1(t)}{W_2(t)} < \frac{\rho_1}{\rho_2} = 1$, and R_2 is $\frac{W_1(t)}{W_2(t)} > 1$. Recalling the definition of r_f, it is easily verified that the region specified in (8.4) corresponds to the darkly shaded regions in Figure 4.

In order to develop an explicit description of the release region, an explicit expression is required for the $(I-1)$–dimensional bounded region that characterizes the solution to the constrained singular control problem. However, only a numerical solution to the constrained singular control problem has been derived in this paper. Thus, the numerical solution needs to be transformed into an explicit expression for the bounded region. In Section 10, an example is carried out for a three-station network, and the reflecting bound-

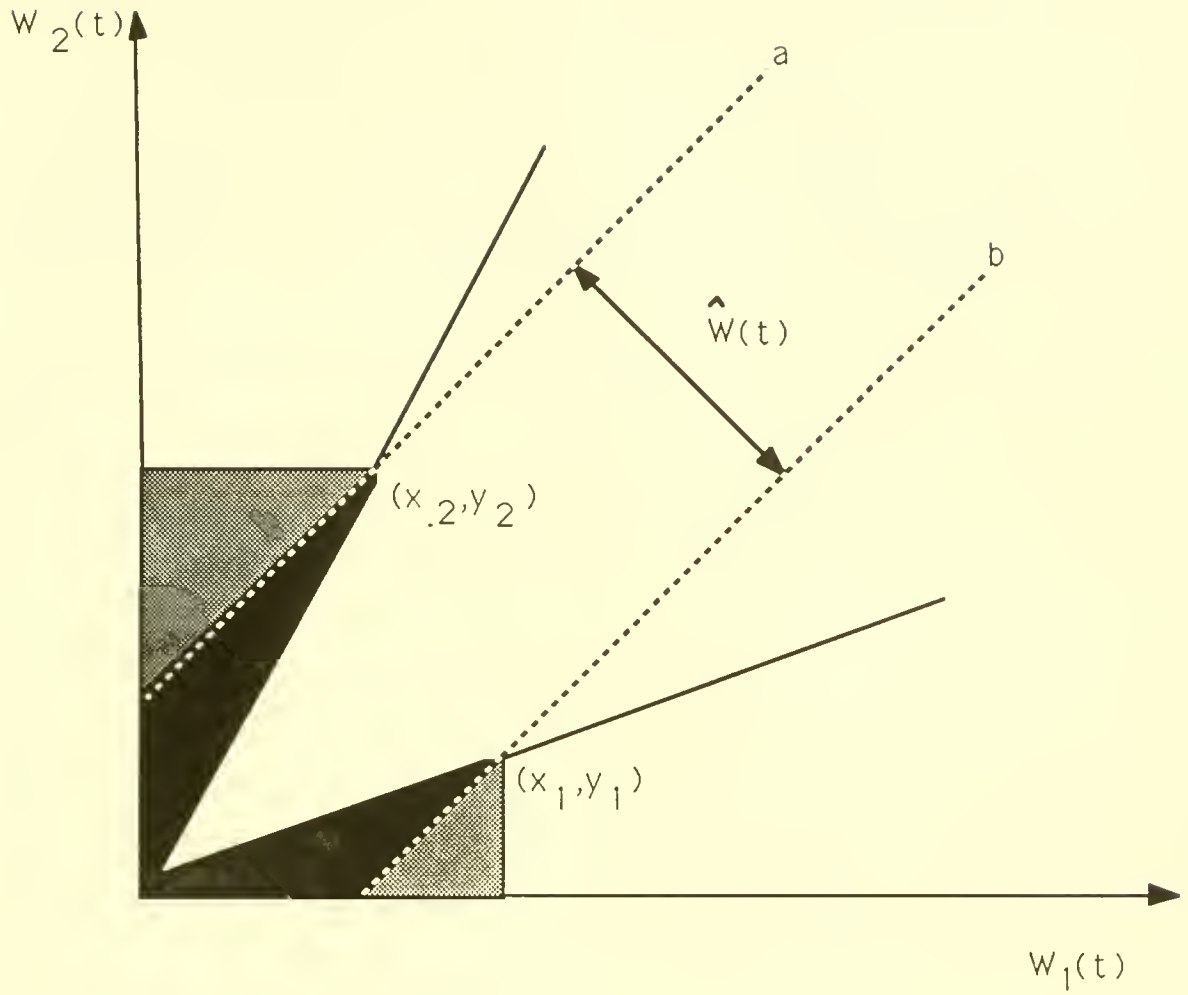

FIGURE 4. The Release Region in a Two-Station Example.

ary containing the bounded region in R^2 is approximated by a piecewise linear boundary in order to develop an explicit release region in R^3. Presumably, an approximation in this spirit is required to develop release regions when $I > 3$.

Notice that the workload regulating input policy defined above would ignore a difference that exists between the actual queueing system and the idealized heavy traffic limit. As pointed out in Wein [21], this difference can be understood by making the following observation about Figure 4. In the idealized Brownian setting, when the scaled workload process W is on the lower ray of the cone boundary and $W_1(t) < x_1$, then there are

zero scaled customers at station 2 and yet *station 2 is not idle*. This apparent paradox is due to the rescaling that occurs when passing to the heavy traffic limit. In the actual queueing system, there are enough customers at the particular station to avoid idleness, but when looked at in the scaled space of the heavy traffic limit, these customers vanish. This difference may prevent the workload regulating input policy to achieve the desired throughput.

However, the release rule can be adapted to the actual queueing system by enlarging the release region. There are two ways this can be achieved. The first way is to slightly enlarge the region on the inside of the cone boundary. This is done by translating the vertex of the cone from $(0, ..., 0)$ to $(\epsilon, ..., \epsilon)$. This translation changes the hyperplane (8.2) to

$$\sum_{i=1}^{I} a_i W_i(t) = \epsilon(\sum_{i=1}^{I} a_i). \tag{8.5}$$

This translation, which may be negligible in scaled space, prevents the process W from straying very far from the original truncated cone boundary. The second way to enlarge the release region is to inflate the bounded region derived in Section 6 by a constant factor $\kappa \geq 1$, while still maintaining the relative shape of the bounded region. As ϵ and κ increase, the servers will incur less idleness but the queue lengths may grow as a result. The workload regulating input policy sets the parameter ϵ and κ so that the desired output rate $\bar{\lambda}$ is achieved. In an actual queueing system, the setting of ϵ and κ will depend upon a variety of factors, including the amount of variability in the queueing system, the amount of time customers spend at non-bottleneck stations, the network topology, and the extent to which the network is heavily loaded.

Notice that the input policy defined in this section has been described in terms of the *scaled* workload process W. Thus, before implementing this policy in an actual queueing system, the release region needs to be expressed in terms of the actual (unscaled) workload process, which is denoted by $w(t) = \{(w_1(t), ..., w_I(t)), t \geq 0\}$. Then $w_i(t) = \sum_{k=1}^{K} M_{ik} Q_k(t)$ for $i = 1, ..., I$ and $t \geq 0$, where $Q_k(t)$ is the actual number of

class k customers in the system at time t. By equation (2.2) it follows that

$$W_i(t) = \frac{w_i(nt)}{\sqrt{n}} \text{ for } i = 1, ..., I \text{ and } t \geq 0. \tag{8.6}$$

Since the scheduling problem was solved using the long-run average criterion, the time scaling can be ignored, and replacing $W_i(t)$ by $w_i(t)/\sqrt{n}$ in the inequalities defining the release region (8.4) will lead to an implementable release policy for the original queueing network.

9. The Workload Balancing Input Heuristic

The customer release policy described in the last section allows for the controller to decide when to release the next customer into the system, but not to choose the class of the entering customer. It was assumed that the class designations of entering customers are exogenously chosen so that q_k is the long-run proportion of class k customers released into the network. In this section, we develop a *workload balancing* input heuristic, which is based on insight gained from the solution to the constrained singular control problem, that decides which class of customer to release into the system. This scheme appears to improve the performance of the scheduling policy and is guaranteed to keep the actual mix of released customers sufficiently close to the desired mix q.

The key idea behind the heuristic is the observation that server idleness is incurred in the idealized Brownian network only when the $(I-1)-$dimensional workload imbalance process $\hat{W}$ reaches the reflecting boundary derived in Section 6. The workload imbalance process stays within a region containing the origin, but when it reaches the reflecting boundary, the workload becomes too imbalanced, the control U is exerted, and at least one server incurs idleness.

Thus, server idleness would be reduced, and hence system performance would be improved, if the workload imbalance process could be discouraged from reaching the reflecting

boundaries. Recall by equations (3.13) and (7.2) that

$$\hat{W}_i(t) = \sum_{k=1}^{K} \hat{M}_{ik} Z_k(t) \text{ for } i = 1, ..., I-1 \text{ and } t \geq 0. \tag{9.1}$$

This equation relates the workload imbalance process $\hat{W}$ to the vector queue length process Z. Our heuristic will release the customer class k that attempts to *balance* the workload imbalance process and hence avoid server idleness.

Consider the actual (unscaled) queueing system, where $\hat{w}_i(t)$ is the actual workload imbalance process at time t and $Q_k(t)$ is the actual number of class k customers in the system at time t; then $\hat{w}_i(t) = \sum_{k=1}^{K} \hat{M}_{ik} Q_k(t)$ for $i = 1, ..., I-1$ and $t \geq 0$. Suppose the input policy derived in the last section dictates that a customer is to be released into the system at time t. There are two steps in the workload balancing input heuristic. The first step checks to see if the actual mix of customers already released into the network is sufficiently close to the derived mix q. Let $N_k(t)$ denote the total number of class k customers released into the system in the time interval $[0, t]$. Let $E = \{k = 1, ..., K | q_k > 0\}$ be the set of possible entering classes; these classes correspond to the first stage of some customer type's route. Consider the constraints

$$q_j N_j(t) - q_k N_k(t) \leq N^* \text{ for all } j, k \in E, \tag{9.2}$$

where N^* is an exogenously specified parameter that specifies how close the actual entering class mix must stay to the target mix q.

If any of the constraints in (9.2) are violated, then the heuristic releases a class l customer, where

$$\max_{j,k \in E} \{q_j N_j(t) - q_k N_k(t)\} = q_m N_m(t) - q_l N_l(t). \tag{9.3}$$

That is, we release the customer class that is lagging behind the farthest from its desired target. If constraints (9.2) are all satisified, then the actual mix of released customers is sufficiently close to the desired mix, and we can proceed to the second step of the heuristic, which attempts to balance the workload.

Let $\bar{\hat{w}}_i(t)$ equal the time average value of $\hat{w}_i$ over the time interval $[0,t]$. This value can be easily calculated in a computer simulation model of a queueing network or in an actual factory that is under computer control; if the factory is not under computer control, then the value of $\hat{w}_i(t)$ may be recorded periodically, for example once per shift, and $\bar{\hat{w}}_i(t)$ can be updated accordingly. Let $\hat{w}_{ik}(t)$ be defined by

$$\hat{w}_{ik}(t) = \hat{w}_i(t) + M_{ik} \text{ for } i = 1, ..., I-1 \text{ and } k \in E. \tag{9.4}$$

Thus, $\hat{w}_{ik}(t)$ is the ith component of the workload imbalance process that would result if a class k customer were released into the system at time t. Step two of the workload balancing input heuristic releases a class l customer into the network, where

$$\min_{k \in E} \{ \sum_{i=1}^{I} \big(\hat{w}_{ik}(t) - \bar{\hat{w}}_i(t) \big)^2 \} = \sum_{i=1}^{I} \big(\hat{w}_{il}(t) - \bar{\hat{w}}_i(t) \big)^2. \tag{9.5}$$

Thus, step two attempts to push the workload imbalance process toward its long-run average value, which presumably will not be close to the reflecting boundary. Notice that the time-average value of $\hat{w}$ is chosen as the desired target, as opposed to choosing the origin (that is, the point $(0, ..., 0)$) as the target. This is because, depending on the workload profile matrix M and the topology of the network, it is possible that the origin will not be a particularly desirable value of $\hat{w}$, in terms of avoiding server idleness.

The workload balancing input heuristic is equally applicable to multiclass closed queueing networks (see Section 4 of Harrison and Wein [8]) because the same relationship holds between server idleness and the workload imbalance process. In a closed network, a new customer is released into the network whenever a customer exits, and this heuristic can be used to decide which class of customer to release. This heuristic was tested on a simulation model of the two-station closed network example in Section 6 of Harrison and Wein [8]. The example there had two customer types, denoted by A and B, and the desired input mix was 50-50. The workload balancing input heuristic offered a 7.8% improvement in average cycle time (from 54.9 to 50.6; see Table I of [8]) over deterministic input (releasing customers in the order ABABAB...), while maintaining the same average throughput

rate. Similarly, the heuristic offered a 10.1% reduction in average cycle time (from 38.6 to 34.7) for the controllable input case (see Table I of Wein [21]). For both of these cases, the exogenous parameter N^* was chosen to guarantee that the resulting class mix would be within one-half of 1% of the desired 50-50 mix.

Furthermore, this simple heuristic is probably effective for any factory, regardless of the timing of its input policy (exogenous, closed, or controllable) or priority sequencing policy. The $(I-1)$–dimensional workload imbalance process offers a concise and effective measure of the balance of work throughout a complex network, and its crucial relationship to the server idleness process is exploited by this heuristic.

10. An Example

The procedure described in this paper will be illustrated by means of a three-station example. The example has three customer types, denoted by A, B, and C, and the specified product mix is to have equal numbers of all three types. Table I describes the deterministic route of each customer type, and gives the mean processing time (in arbitrary time units) for each of the various stages of service. All service time distributions are assumed to be exponential, although our results hold for any service time distribution with finite mean and variance.

CUSTOMER TYPE	ROUTE	MEAN SERVICE TIMES
A	$3 \rightarrow 1 \rightarrow 2$	6.0 4.0 1.0
B	$1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2$	8.0 6.0 1.0 2.0 7.0
C	$2 \rightarrow 3 \rightarrow 1 \rightarrow 3$	4.0 9.0 4.0 2.0

Since each customer class corresponds to a type-stage pair, the twelve customer classes

are designated (and ordered from $k = 1, ..., 12$) by (A1,A2,A3,B1,...,B5,C1,...,C4). The 12×12 routing matrix P has non-zero entries $P_{12} = P_{23} = P_{45} = P_{56} = P_{67} = P_{78} = P_{9,10} = P_{10,11} = P_{11,12} = 1$. Calculation of the 3×12 workload profile matrix M yields

$$M = \begin{pmatrix} 4 & 4 & 0 & 10 & 2 & 2 & 2 & 0 & 4 & 4 & 4 & 0 \\ 1 & 1 & 1 & 13 & 13 & 7 & 7 & 7 & 4 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 11 & 11 & 2 & 2 \end{pmatrix}, \tag{10.1}$$

where M_{ik} is the expected remaining processing time at station i for a class k customer until that customer exits the network. Since $q = (\frac{1}{3}\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0)^T$, equation (2.12) yields $v_1 = v_2 = v_3 = 6$, thus implying perfect system balance by equation (2.13). Therefore, the ρ values can be factored out of equation (7.2), as mentioned in Section 4, and the 2×12 workload imbalance profile matrix $\hat{M}$ can be given by

$$\hat{M} = \begin{pmatrix} -2 & 4 & 0 & 9 & 1 & 1 & 2 & 0 & -7 & -7 & 2 & -2 \\ -5 & 1 & 1 & 12 & 12 & 6 & 7 & 7 & -7 & -11 & -2 & -2 \end{pmatrix}. \tag{10.2}$$

The exogenous output rate is .15 customers per unit of time, so that, by (2.13), $\rho_1 = \rho_2 = \rho_3 = .9$, and the system parameter value of $n = 100$ can be chosen. The holding costs are $c_k = 1$ for $k = 1, ..., 12$, so that the objective is to minimize the long-run expected average number of customers in the system (or the long-run expected average cycle time) subject to meeting the long-run expected average output rate of .15 customers per unit time.

The dual LP (3.9)-(3.10) can be expressed as

$$\max_{\pi_1(t), \pi_2(t)} \hat{W}_1(t)\pi_1(t) + \hat{W}_2(t)\pi_2(t) \tag{10.3}$$

$$\text{subject to } \hat{M}^T \pi(t) \leq e, \tag{10.4}$$

where $\hat{M}$ is given in (10.2). This problem can be solved graphically for all values of the workload imbalance process $\hat{W}$, and the solution is given in Table II. There are six extreme points of the constraint set (10.4), and thus the scaled workload process W lives on a polyhedral cone with six faces. The six extreme points lead to the workload imbalance space R^2 being partitioned into six regions, which were referred to in Section 8 as R_f,

and which are numbered in Table II for future reference. For each of the six regions, dynamic reduced costs are calculated according to (3.11) and a priority sequencing policy is developed according to rules (7.1)-(7.3). The resulting sequencing policy is given in Table III.

In order to find the workload regulating input policy, a numerical solution is needed to problem (4.24)-(4.26). The objective function cost $h(\hat{W}(t))$ defined in (4.2) is found from $\pi^*(t)$ in Table II. By (2.14), the righthand side values in (4.25) are $\gamma_1 = \gamma_2 = \gamma_3 = 1$. By (4.4), the drift of the two-dimensional Brownian motion process $\hat{B}$ is (0,0) and the calculations in (2.7), (2.17), (2.18), and (4.3) yield the covariance matrix

$$a = \begin{pmatrix} 12.333 & 6.778 \\ 6.778 & 12.444 \end{pmatrix}. \tag{10.5}$$

REGION NUMBER AND DESCRIPTION	DUAL SOLUTION
1: $\hat{W}_1(t) > 0, \hat{W}_2(t) > 0, \frac{3}{4} \leq \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq 4;$	$\pi_1^*(t) = \frac{11}{39}, \pi_2^*(t) = -\frac{5}{39}$
2: $\hat{W}_1(t) > 0, \hat{W}_2(t) > 0, \frac{1}{12} \leq \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq \frac{3}{4};$	$\pi_1^*(t) = 0, \pi_2^*(t) = \frac{1}{12}$
3: $\hat{W}_1(t) \leq 0, \hat{W}_2(t) > 0;$	
$\hat{W}_1(t) < 0, \hat{W}_2(t) = 0;$	$\pi_1^*(t) = -\frac{19}{77}, \pi_2^*(t) = \frac{8}{77}$
$\hat{W}_1(t) > 0, \hat{W}_2(t) > 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq \frac{1}{12};$	
$\hat{W}_1(t) < 0, \hat{W}_2(t) < 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \geq 1;$	
4: $\hat{W}_1(t) < 0, \hat{W}_2(t) < 0, \frac{7}{11} \leq \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq 1;$	$\pi_1^*(t) = -\frac{1}{7}, \pi_2^*(t) = 0$
5: $\hat{W}_1(t) = 0, \hat{W}_2(t) < 0;$	
$\hat{W}_1(t) < 0, \hat{W}_2(t) < 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq \frac{7}{11};$	$\pi_1^*(t) = \frac{1}{4}, \pi_2^*(t) = -\frac{1}{4}$
$\hat{W}_1(t) > 0, \hat{W}_2(t) < 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \geq -1;$	
6: $\hat{W}_1(t) > 0, \hat{W}_2(t) = 0;$	
$\hat{W}_1(t) > 0, \hat{W}_2(t) > 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \geq 4;$	$\pi_1^*(t) = \frac{3}{10}, \pi_2^*(t) = -\frac{2}{10}$
$\hat{W}_1(t) > 0, \hat{W}_2(t) < 0, \frac{\hat{W}_1(t)}{\hat{W}_2(t)} \leq -1;$	

TABLE II. Dual Solution as a Function of Workload Imbalance

Using a mesh size of $h = 0.5$ in the finite difference approximation, the solution to the LP (6.1)-(6.4), (6.8) gives the reflecting boundary shown in Figure 5. Superimposed on top of this boundary is the partition of the six regions $R_f, f = 1, ..., 6$, corresponding to the six dual LP solutions described in Table II. The workload imbalance process is uncontrolled when it is strictly within the reflecting boundary, and when it reaches the boundary, the process is pushed back in (in the direction of the arrows in Figure 5) by at least one of the three controls (U_1^*, U_2^*, U_3^*). Notice that there are places on the boundary where more than one control is used at a given time. In particular, U_1^* and U_3^* are used at states (0.5,9.5), (1,9.5), and (1.5,10), and U_2^* and U_3^* are both used when the process reaches (3,0), (3,0.5), (3,1), and (3,1.5).

REGION	STATION 1	STATION 2	STATION 3
1	B4 C3 A2 B1	A3 B5 B2 C1	C4 B3 C2 A1
2	C3 A2 B4 B1	A3 C1 B5 B2	C4 A1 C2 B3
3	A2 C3 B4 B1	A3 B5 C1 B2	C4 A1 B3 C2
4	A2 C3 B4 B1	A3 B5 B2 C1	C4 B3 A1 C2
5	B4 B1 A2 C3	A3 B5 B2 C1	C4 B3 A1 C2
6	B4 B1 C3 A2	A3 B5 B2 C1	C4 B3 A1 C2

TABLE III. Priority Sequencing Policy

As mentioned in Section 8, the numerical solution to the constrained singular control problem needs to be transformed into an explicit expression for the reflecting boundary. A piecewise linear boundary consisting of 13 segments (see Figure 6) was used to approximate the reflecting boundary from Figure 5. The release region (8.4) for this example is given in terms of the actual (unscaled) workload process w in the Appendix; the bounded region R_B is expressed in the Appendix in terms of the workload process rather than the workload imbalance process. Also, recall that the multiplicative factor κ appearing in the Appendix inflates the bounded region in Figure 6.

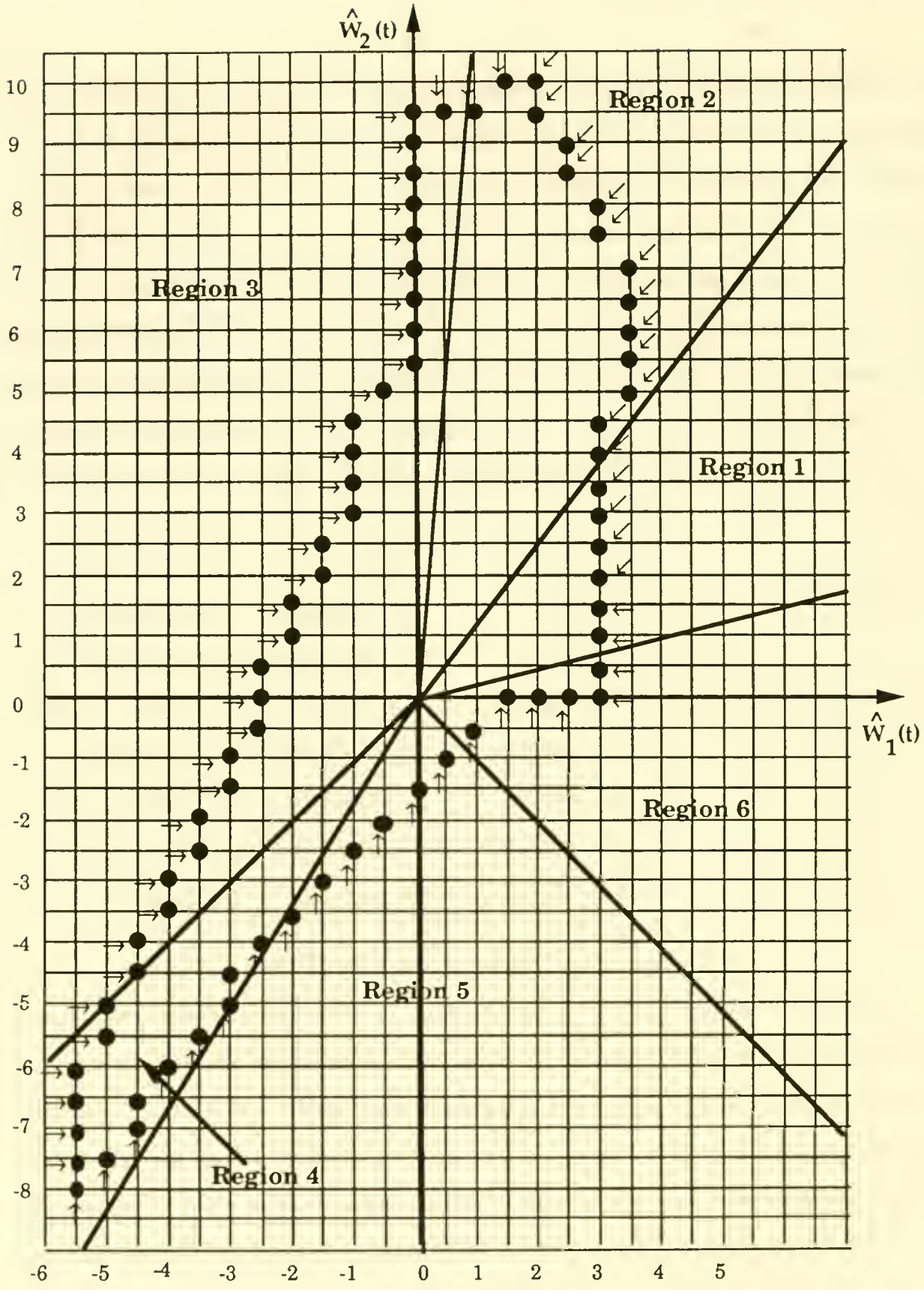

FIGURE 5. The Reflecting Boundary in the Constrained Singular Control Problem.

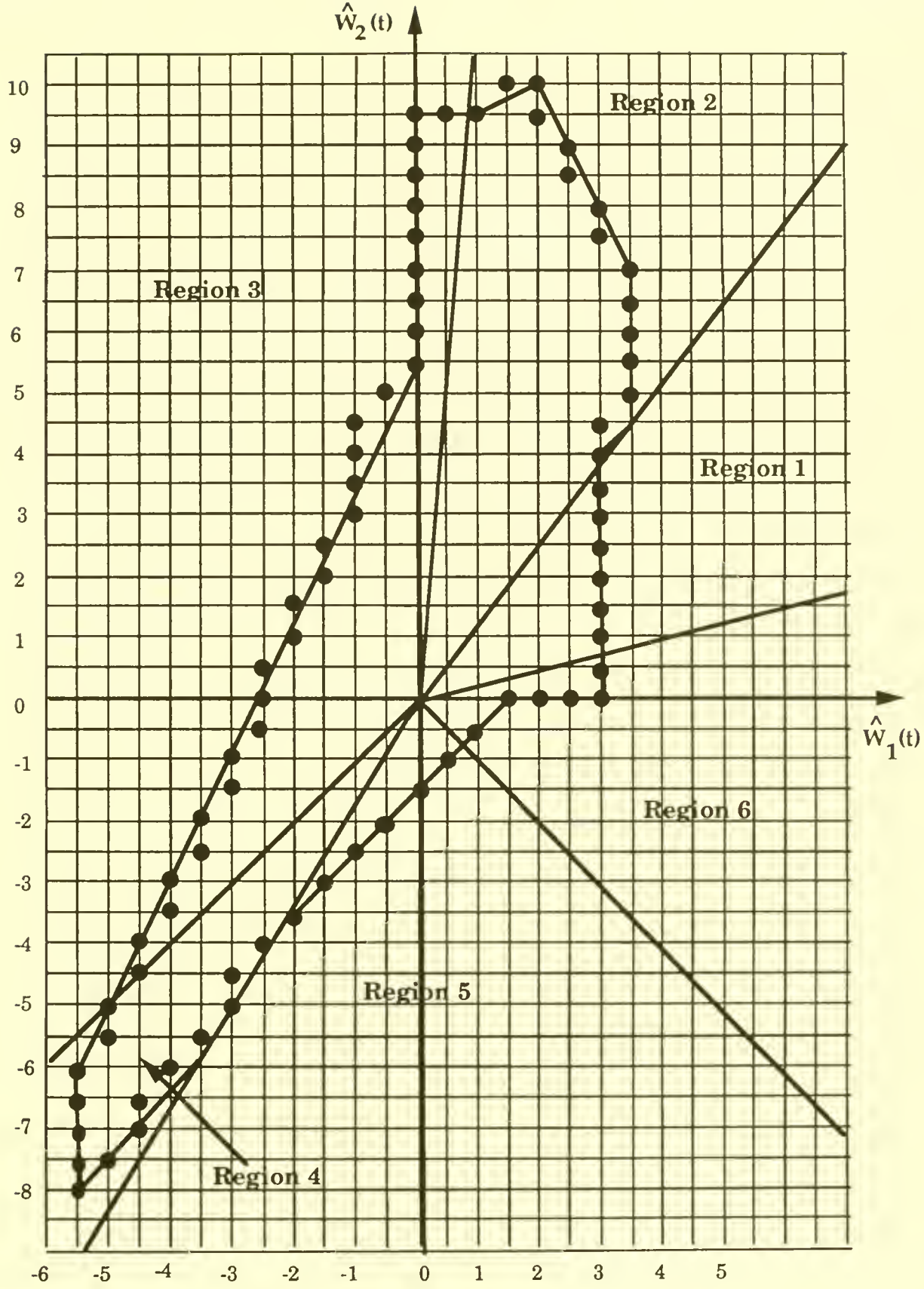

FIGURE 6. A Piecewise Linear Approximation to the Reflecting Boundary.

Using the example network, a simulation study was undertaken to compare the performance of the scheduling policies proposed in Sections 8, 9, and 10 against several conventional customer release and priority sequencing policies. Two conventional input policies were tested: deterministic, where the interarrival times are all constant, and closed loop input, where the total number of customers in the network is held constant at N; the latter policy is abbreviated by CLOSED(N) in Table IV. For both of these input rules, customers were released into the network in the order ABCABCABC... Both input policies were tested in conjunction with two priority sequencing rules: first-in first-out (FIFO) and the shortest expected remaining processing time rule (SERPT), where priority is given to the customer class k with the smallest value of $\sum_{i=1}^{I} M_{ik}$.

The scheduling policy proposed here is abbreviated in Table IV by WR(ϵ, κ), WBAL(N^*), and DRC (for dynamic reduced costs), where the workload regulating input policy with parameters ϵ and κ dictates via (8.4) when a customer is to be released, the workload balancing input heuristic with parameter N^* dictates the class of customer to be released, and the priority sequencing policy is defined by the dynamic indices in (7.1) and (7.3).

INPUT RULE	SEQUENCING RULE	THROUGHPUT RATE (95% C.I.)	CYCLE TIME (95% C.I.)
DETERMINISTIC	FIFO	.149(±.0001)	144(±10.4)
DETERMINISTIC	SERPT	.149(±.0002)	182(±15.7)
CLOSED(18)	FIFO	.149(±.0010)	120(±0.8)
CLOSED(25)	SERPT	.149(±.0008)	166(±1.1)
WR(1.7,1.5),WBAL($\frac{16}{3}$)	DRC	.149(±.0008)	85.4(±1.1)

TABLE IV. Comparison of Cycle Times

The results of the simulation study are summarized in Table IV. Each row gives

statistics for a particular scheduling policy, which is a combination of a customer release policy and a priority sequencing rule. For each policy tested, 20 independent runs were made, each consisting of 5000 customer completions. The first 200 time units of each run was truncated to reduce the initial bias. The third and fourth columns give the average throughput rate and average cycle time, respectively, over the 20 runs, along with 95% confidence intervals. The parameters N, ϵ, and κ were chosen so that all scheduling policies achieved the average throughput rate of .149 customer completions per unit time, which corresponds to a server utilization of 89.4%. Recall that the objective is to minimize the average cycle time subject to a given average throughput rate.

Referring to Table IV, it is seen that the scheduling policy proposed in this paper easily outperforms all of the conventional scheduling rules. The policy offers a 28.8% reduction in average cycle time over the (CLOSED,FIFO) case, which was its closest competitor. It also achieved a 40.7% reduction relative to the (DETERMINISTIC,FIFO) case, even though it is well known (see Whitt [23], for example) that reducing the variability in the interarrival times of a queueing network will lead to improved performance.

The parameter value $\epsilon = 1.5$ corresponds to 15.0 units of unscaled work, which in turn roughly corresponds to the amount of work for each server that is embodied in two and one-half customers, since $v_i = 6$ for $i = 1, 2, 3$. The parameter value $\kappa = 1.7$ means that the bounded region in Figure 6 was enlarged by 70%. The parameter value $N^* = \frac{16}{3}$ guaranteed that the actual entering fraction of each customer class was within .003 of the specified target of $q_k = .333$; in the simulation study, the resulting class mix was virtually equal to the target mix.

11. Conclusions

In this paper we have considered the problem of how to dynamically release jobs

(when and which class) and prioritize jobs in a multistation multiclass queueing network with general service time distributions and a general routing structure. The objective was to minimize the long-run expected average linear holding costs of customers, subject to a specified average input mix and a constraint on the long-run expected average output rate. Under balanced heavy loading conditions, the scheduling problem was approximated by a Brownian control problem, and a numerical solution to the workload formulation of the control problem was obtained.

This solution was then interpreted in terms of the original queueing system in order to develop an effective three-part scheduling policy. The first part is the workload regulating input policy, which releases a job whenever the workload process enters a particular region. The second part is the workload balancing input heuristic, which releases the customer class that will best balance the workload among the various bottleneck stations. The third part is the priority sequencing policy, which assigns dynamic indices (based on dynamic reduced costs from a linear program) to each customer class. A computational study was performed that exhibited the policy's effectiveness.

Two related research topics are to prove a weak convergence result for the finite difference approximation used in Section 5, and to develop an efficient algorithm to solve the large linear program posed in Section 6. Also, the close relationship between the problem addressed in this paper and the problem of priority sequencing in a multistation multiclass closed queueing network requires further investigation. Finally, more numerical studies need to be performed to better understand the behavior and robustness of the proposed scheduling policy.

Appendix

Let the regions $R_f, f = 1, ..., 6$ be defined as in the first column of Table II (with $W(t)$

replaced by $w(t)$). The release region (8.4) for the example in Section 10 is to release a customer whenever the workload process $w(t)$ enters $\cap_{1\le f\le 6}\bar{R}_f$, where $\bar{R}_f = R_f \cap S_f$, and:

S_1 : $w_1(t) - 4w_2(t) + 42w_3(t) \le 390\epsilon$, and
$w_1(t) - w_3(t) \le 30\kappa$.

S_2 : $-w_2(t) + 13w_3(t) \le 120\epsilon$,
$w_1(t) - w_3(t) \le 35\kappa$,
$2w_1(t) + w_2(t) - 3w_3(t) \le 140\kappa$, and
$-w_1(t) + 2w_2(t) - w_3(t) \le 180\kappa$.

S_3 : $139w_1(t) - 18w_2(t) - 44w_3(t) \le 770\epsilon$,
$w_2(t) - w_3(t) \le 95\kappa$, and
$-w_1(t) + w_3(t) \le 0$;
or
$139w_1(t) - 18w_2(t) - 44w_3(t) \le 770\epsilon$, and
$-11w_1(t) + 5w_2(t) + 6w_3(t) \le 300\kappa$.

S_4 : $11w_1(t) - 4w_3(t) \le 70\epsilon$,
$-w_1(t) + w_3(t) \le 55\kappa$, and
$w_1(t) - w_2(t) \le 25\kappa$.

S_5 : $w_2(t) \le 10\epsilon$, and
$w_1(t) - w_2(t) \le 15\kappa$.

S_6 : $-w_1(t) + 4w_2(t) + 2w_3(t) \le 50\epsilon$, and
$w_1(t) - w_2(t) \le 15\kappa$;
or
$-w_1(t) + 4w_2(t) + 2w_3(t) \le 50\epsilon$,
$-w_2(t) + w_3(t) \le 0$, and

$$w_1(t) - w_3(t) \leq 30\kappa.$$

Acknowledgements

This research was partially supported by a Junior Faculty Grant from the Leaders for Manufacturing Program at MIT. I would like to thank Phillipe Chevalier for his invaluable assistance with the computations in Section 10 and for his helpful comments pertaining to the material in Section 8, and Robert Freund for drawing Figure 1.

REFERENCES

[1] Beneš, V. E., Shepp, L. A. and Witsenhausen, H. S. (1980). Some Solvable Stochastic Control Problems. *Stochastics* 4, 39-83.

[2] Chen, H. and Mandelbaum, A. (1987). Stochastic Flow Networks: Bottlenecks and Diffusion Approximations. Preprint, Graduate School of Business, Stanford U., Stanford, Ca.

[3] Chow, P. L., Menaldi, J. L., and M. Robin (1985). Additive Control of Stochastic Linear Systems With Finite Horizon. *SIAM J. Control and Opt.* 23, 858-899.

[4] Derman, C. (1966). Denumerable State Markov Decision Processes - Average Criterion. *Ann. Math. Statist.* 37,1545-1554.

[5] Harrison, J. M. (1973). A Limit Theorem for Priority Queues in Heavy Traffic. *J. Appl. Prob.* 10, 907-912.

[6] Harrison, J. M. (1985). *Brownian Motion and Stochastic Flow Systems.* John Wiley and Sons, New York.

[7] Harrison, J. M. (1988). Brownian Models of Queueing Networks with Heterogeneous Customer Populations, in W. Fleming and P. L. Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume 10, Springer-Verlag, New York, 147-186.

[8] Harrison, J. M. and Wein, L. M. (1988). Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network. To appear in *Operations Research.*

[9] Johnson, D. P. (1983). Diffusion Approximations for Optimal Filtering of Jump Processes and for Queueing Networks. Unpublished Ph.D. thesis, Dept. of Mathematics, Univ. of Wisconsin, Madison.

[10] Karatzas, I. (1988). Stochastic Control Under Finite-fuel Constraints, in W. Fleming and P. L. Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume 10, Springer-Verlag, New York, 225-240.

[11] Klimov, G. P. (1974). Time Sharing Service Systems I. *Th. Prob. Appl.* 19,532-551.

[12] Kushner, H. J. (1977). *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.* Academic Press, New York.

[13] Kushner, H. J. (1988). Numerical Methods for Stochastic Control Problems in Continuous Time. Technical Report, Div. Applied Math., Brown U., Providence, R. I.

[14] Kushner, H. J. and Chen, C. H. (1974). Decomposition of Systems Governed by Markov Chains. *IEEE Transactions Aut. Cont.* AC-19, 501-507.

[15] Manne, A. (1960). Linear Programming and Sequential Decisions. *Management Science* 6, 259-267.

[16] Menaldi, J. L., Robin, M., Taksar, M. I. (1988). Singular Ergodic Control for Multidimensional Gaussian Processes. Submitted for publication.

[17] Peterson, W. P. (1985). Diffusion Approximations for Networks of Queues with Multiple Customer Types. Unpublished Ph.D. Thesis, Dept. of Operations Research, Stanford University.

[18] Reiman, M. I. (1983). Some Diffusion Approximations with State Space Collapse. *Proc. Intl. Seminar on Modeling and Performance Evaluation Methodology*, Springer-Verlag, Berlin.

[19] Taksar, M. I. (1985). Average Optimal Singular Control and a Related Stopping Problem. *Math. of Operations Research* 10, 63-81.

[20] Wein, L. M. (1988). Optimal Control of a Two-Station Brownian Network. To appear in *Math. of Operations Research.*

[21] Wein, L. M. (1988). Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Network With Controllable Inputs. To appear in *Operations Research.*

[22] Whitt, W. (1971). Weak Convergence Theorems for Priority Queues: Preemptive-Resume Discipline. *J. Appl. Prob.* 8, 74-94.

[23] Whitt, W. (1984). Open and Closed Models for Networks of Queues. *AT&T Bell Laboratories Technical Journal* 63, 1911-1979.

[24] P. Yang, Pathwise Solutions for a Class of Linear Stochastic Systems, Unpublished Ph. D. Thesis, Dept. of Operations Research, Stanford University, Stanford, CA, 1988.