Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Closed Network

Philippe B. Chevalier and Lawrence M. Wein

OR 219-90 July, 1990

SCHEDULING NETWORKS OF QUEUES: HEAVY TRAFFIC ANALYSIS OF A MULTISTATION CLOSED NETWORK

Philippe B. Chevalier

Operations Research Center, M.I.T.

and

Lawrence M. Wein

Sloan School of Management, M.I.T.

Abstract

We consider the problem of finding an optimal dynamic priority sequencing policy to maximize the mean throughput rate in a multistation, multiclass closed queueing network with general service time distributions and a general routing structure. Under balanced heavy loading conditions, this scheduling problem can be approximated by a control problem involving Brownian motion. Although a unique, closed form solution to the Brownian control problem is not derived, an analysis of the problem leads to an effective static sequencing policy, and to an approximate means of comparing the relative performance of arbitrary static policies. Three examples are given that illustrate the effectiveness of our procedure.

July 1990

SCHEDULING NETWORKS OF QUEUES: HEAVY TRAFFIC ANALYSIS OF A MULTISTATION CLOSED NETWORK

Philippe B. Chevalier

Operations Research Center, M.I.T.

and

Lawrence M. Wein

Sloan School of Management, M.I.T.

Multiclass closed queueing networks are important models for computer, communication, and manufacturing systems, and the descriptive theory of these networks is well developed (see Baskett et al. 1975 and Kelly 1979). However, no exact results exist for optimal priority sequencing in such systems, and the only approximate analysis is Harrison and Wein (1990), who obtain an effective priority sequencing policy for maximizing the throughput of a two-station, well balanced, heavily loaded network. This policy, called a *workload balancing* sequencing policy, is a static (that is, not state-dependent) policy that outperformed conventional sequencing policies in a simulation study in Harrison and Wein (1990). This result was obtained by analyzing a Brownian system model (developed by Harrison 1988) 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. The workload balancing sequencing policy was derived by reformulating the Brownian control problem in terms of workload imbalances, solving the workload imbalance formulation, and interpreting the solution in terms of the original queueing system.

In this paper, we attempt to generalize the results of Harrison and Wein (1990) from the setting of a two-station network to a network with any finite number of stations. In order to describe our results, it is easiest to first review the results of Harrison and

Wein (1990). They define a one-dimensional *workload imbalance* process, which measures how imbalanced the total network workload is between the two stations at each point in time, and discover an intricate relationship between workload imbalance, server idleness, and the lowest priority customer classes. In particular, in the idealized Brownian limit, server idleness is only incurred at times when the workload imbalance process is on the boundary of a *workload imbalance polytope*, which is a closed interval on the real line, and the two extreme points of the polytope correspond to the two customer classes, one from each station, that are awarded lowest priority at their respective stations. These two bottom priority classes, which are referred to as *extremal classes*, lead directly to the workload balancing sequencing policy. Furthermore, this relationship allows for an analytic comparison between the workload balancing policy and any other static policy, such as the shortest expected processing time rule (SEPT), where priority is given to the class with the shortest expected processing time for its upcoming operation, and the shortest expected remaining processing time rule (SERPT), where priority is given to the class with the least expected amount of work remaining before exiting the network.

For the general multistation problem considered here, the Brownian control problem can again be reformulated in terms of workload imbalances, but a unique, closed form solution to the workload imbalance formulation is not obtained. However, the corresponding relationship between workload imbalance, server idleness, and the lowest priority classes is generalized to the multistation setting. In particular, when there are I stations in the network, an $(I-1)-$dimensional workload imbalance process is defined that stays in a workload imbalance polytope in R^{I-1}. Also, server idleness is incurred only when the workload imbalance process is at the boundary of the workload imbalance polytope. Each extreme point of the polytope corresponds to a particular customer class, and these extremal classes are the only classes in the network that are ever given bottom priority at their respective stations. Unlike the two-station case, there will in general be more extremal classes than stations.

The insight gained from the previous paragraph allows us to identify an effective static sequencing policy for maximizing the throughput of a multistation, multiclass closed queueing network under balanced heavy loading conditions, and to approximately compare the performance of this policy to conventional static policies, such as the SEPT and SERPT rules. A simulation study analyzing three examples (two three-station networks and a four-station network) are carried out that demonstrate the power of the simple procedure of identifying the workload imbalance polytope and the corresponding extremal classes. In particular, for each example, the proposed policy easily (and at times, dramatically) outperforms four conventional policies, and the analysis roughly predicts the relative performance of the proposed policy, the SEPT rule, and the SERPT rule. Also, system performance is greatly influenced by operating under different static policies.

Perhaps the most interesting conclusion of our study is the effectiveness of *static* policies for maximizing the throughput in multistation closed queueing networks. In contrast, when analyzing perhaps the simplest interesting open queueing network scheduling problem, Harrison and Wein (1989) found that no static policy was effective, and a dynamic (that is, state-dependent) policy was required to offer significant improvement over the first-come first-served (FCFS) policy. We believe this is due to the fundamental tradeoff that exists in all open queueing networks. This tradeoff is between the short run aim of reducing the number of customers in the system, and the longer run aim of avoiding server idleness. On the other hand, in a single-station queue, no such tradeoff exists, and the only concern is with reducing the number of customers in the system. Therefore, it is not surprising that a simple static policy (the so-called $c\mu-$rule; see Klimov 1974, for example) is able to achieve this goal. Similarly, no tradeoff exists in a closed network setting, where server utilization is the sole concern, and so a static policy again appears to be effective, although obviously not optimal. In summary, it appears that the basic tradeoff that exists in sequencing open networks makes these systems more difficult to analyze and to sequence than closed networks or single-station systems.

The balanced heavy loading conditions imply that any stations in the original network that are not among the most heavily loaded will vanish in our idealized Brownian model. Thus, the proposed sequencing policy can be applied to any closed queueing network by restricting attention to the subnetwork of bottleneck stations. Although our procedure works very well on the bottleneck subnetwork, further study is required to assess the effectiveness of this procedure for scheduling an entire network consisting of bottleneck and nonbottleneck stations. However, the bottleneck stations are precisely where most of the congestion occurs, and where scheduling will have its biggest impact. Thus, we believe these results have the potential to enhance system performance in actual closed network settings.

This paper is organized as follows. In Section 1, the queueing network scheduling problem is described, and the workload imbalance formulation of the approximating Brownian control problem is given in Section 2. The workload imbalance polytope is defined in Section 3, where the relationship between server idleness, workload imbalance, and extremal classes is described. A static sequencing policy is proposed in Section 4, which also contains an approximate analytic comparison between the proposed policy and any other static policy. Three examples are contained in Section 5, along with simulation results.

1. The Queueing Network Scheduling Problem

Consider a queueing network consisting of I single server stations, and populated by a variety of different customer *types*, where each type has its own arbitrary, deterministic route through the network. As in Kelly (1979) and Harrison (1988), we define a different customer *class* for each stage along each customer type's route. Each customer class $k = 1, ..., K$ requires service at a particular station, and has its own general service time distribution with finite mean and variance. Thus, individual customers change class deterministically as they proceed through the network.

Whenever a customer completes the last stage of its route, it exits the network, and a

new customer immediately enters, so as to keep the population size fixed at N customers. The new entering customer will be of class k with probability q_k, independent of all previous history. Of course, $q_k > 0$ only for classes that correspond to the first stage along some customer type's route.

Notice that this is a *single chain* network, where the entering mix of the various customer types is fixed, as opposed to a *multichain* network where the population level of the various customer types is fixed. The single chain network is appropriate for a manufacturing setting, which is our primary interest. In a job shop, the product mix is typically specified by customer demand, and the most direct way to satisfy this mix in a closed network setting is to release new customers according to the appropriate entering class mix $q = (q_k)$. Our results remain unchanged if the class of entering customer is chosen in a deterministic (rather than Markovian) fashion according to the vector q. Also, customer routes are assumed to be deterministic for ease of presentation; probabilistic events that occur in a manufacturing setting, such as rework, scrapping, and server breakdown and repair, can be easily incorporated into the model (see Harrison 1988 for details).

The scheduling problem is to dynamically decide which class of customers to serve next at every station in the network. These decisions will be referred to as *sequencing* decisions. The objective of the scheduling problem is to maximize the long run expected average throughput rate of the network, which is the number of customer departures per unit of time. Since the customer population level is fixed, Little's formula (Little 1961) implies that this objective will also minimize the long run expected average sojourn time of customers, which is the amount of time a customer spends in the network. Since the entering class mix, customer routes, and mean service times are all fixed, maximizing the long run expected average throughput rate is equivalent to minimizing the long run expected average idleness rate for any arbitrary server, which is the fraction of time the server is idle.

2. The Workload Imbalance Formulation

Harrison (1988) has shown how to approximate the closed queueing network scheduling problem described in Section 1 by a Brownian control problem. In Section 2 of Harrison and Wein (1990), an equivalent formulation of this problem is derived for the case $I = 2$. The new formulation is called a *workload formulation* because the state of the queueing system is described in terms of an $I-$dimensional workload process, rather than the $K-$dimensional queue length process. In Section 3 of Harrison and Wein (1990), the workload formulation was easily re-expressed in terms of a *workload imbalance* formulation (see equations (38)-(44) of that paper), where the state of the network is an $(I-1)-$dimensional vector of workload imbalances, which measures how imbalanced the workload of the first $I-1$ stations are relative to station I. Readers are also referred to Wein (1990c) for a similar multidimensional workload imbalance formulation. We will go directly to the workload imbalance formulation of the problem in order to avoid much unnecessary notation. As in Harrison and Wein (1990), the proposed sequencing policy only depends on the solution to the workload imbalance formulation.

Let $Q_k(t)$ be the number of class k customers in the network at time t, and let $I_i(t)$ be the cumulative idleness incurred by server i in the time interval $[0,t]$. The Brownian approximation is obtained by rescaling these two basic processes in terms of the total population size N. In particular, define the scaled *queue length process* $Z_k = \{Z_k(t), t \geq 0\}$ by

$$Z_k(t) = \frac{Q_k(N^2 t)}{N}, \ t \geq 0 \text{ and } k = 1, ..., K, \tag{1}$$

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

$$U_i(t) = \frac{I_i(N^2 t)}{N}, \ t \geq 0 \text{ and } i = 1, ..., I. \tag{2}$$

Notice that $Z_k(t)$ is interpreted as the fraction of customers in the network at time t who are of class k. The vector processes $Z = (Z_k)$ and $U = (U_i)$ are the control processes in the

workload imbalance formulation of the Brownian control problem. For brevity's sake, the scaled processes Z and U will be referred to simply as the queue length and cumulative idleness processes, respectively. Since we will be dealing exclusively with the Brownian model in the next two sections, this should cause no confusion.

Let us define M_{ik} to be the expected remaining processing time at station i for a class k customer until that customer exits the network. The $I \times K$ *workload profile matrix* $M = (M_{ik})$ depends on the mean processing time of each customer class and the detailed route of each customer type. Readers may refer to Table I and equation (16) in Section 5, where the entries of this matrix are displayed for a concrete example.

As mentioned in Section 1, newly injected customers are of class k with probability q_k. For $i = 1, ..., I$, define $v_i = \sum_{k=1}^{K} M_{ik} q_k$, so that v_i is the expected total time over the long run that server i devotes to each newly arriving customer. Recall that in closed queueing networks, the vector of traffic intensities can only be determined up to a scale constant. As in Harrison and Wein (1990), the relative traffic intensitites $\rho = (\rho_i)$ will be scaled so that $\max_{\{1 \le i \le I\}} \rho_i = 1$. By Proposition 2 of Harrison and Wein (1990), it follows that $\rho_i = v_i / \max_{\{1 \le j \le I\}} v_j$, for $i = 1, ..., I$. The *balanced heavy loading conditions* for the closed network assume the existence of a sufficiently large integer N such that the total population size is N and $N|1 - \rho_i|$ is of moderate size for all $i = 1, ..., I$.

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

$$\hat{M}_{ik} = \rho_I M_{ik} - \rho_i M_{Ik} \quad \text{for } i = 1, ..., I-1, \quad \text{and } k = 1, ..., K. \tag{3}$$

As in Harrison and Wein (1990), and Wein (1990b,c), this matrix contains all the necessary information about each customer class to schedule the network under balanced heavy loading conditions.

Let X be a $K-$dimensional Brownian motion process with drift vector δ and covariance matrix Σ, which are defined in equations (13)-(14) of Harrison and Wein (1990) in terms of the first and second moments of the service time distributions of the different

customer classes, the routes of the various customer types, and the entering class mix. Also, let $B = (B_i)$ be defined by $B = TMX$, where the $(I-1) \times I$ matrix T is given by

$$T = \begin{pmatrix} \rho_I & 0 & 0 & . & . & 0 & -\rho_1 \\ 0 & \rho_I & 0 & . & . & . & -\rho_2 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & \rho_I & 0 & -\rho_{I-2} \\ 0 & 0 & 0 & . & 0 & \rho_I & -\rho_{I-1} \end{pmatrix}, \tag{4}$$

so that B is an $(I-1)-$dimensional Brownian motion process with drift $\mu = TM\delta$ and covariance $\Gamma = TM\Sigma M^T T^T$. Although our proposed policies do not depend on the parameter values of these two Brownian motion processes, it is worth noting that the components of the drift vector μ are $\mu_i = N(\rho_i - \rho_I)$ for $i = 1, .., I-1$, by Proposition 3 of Harrison and Wein (1990).

The approximating Brownian control problem is obtained by letting the customer population size $N \to \infty$. By Propositions 2 and 7 of Harrison and Wein (1990), the workload imbalance formulation of the Brownian control problem is to choose RCLL (right continuous with left limits) processes Z and U ($K-$dimensional and $I-$dimensional, respectively) to

$$\text{minimize} \quad \limsup_{t \to \infty} \frac{1}{t} tE[U_1(t)] \tag{5}$$

$$\text{subject to} \quad Z \text{ and } U \text{ are nonanticipating with respect to } X, \tag{6}$$

$$\sum_{k=1}^{K} \hat{M}_{ik} Z_k(t) = B_i(t) + \rho_I U_i(t) - \rho_i U_I(t), \quad \text{for} \quad i = 1, ..., I-1; t \geq 0, \tag{7}$$

$$U \text{ is nondecreasing with } U(0) = 0, \tag{8}$$

$$\sum_{k=1}^{K} Z_k(t) = 1, \quad \text{for all } t \geq 0, \text{ and} \tag{9}$$

$$Z(t) \geq 0, \quad \text{for all } t \geq 0. \tag{10}$$

We conclude this section with several comments on the workload imbalance formulation, which gets its name because the basic system state equation (7) is in terms of the

$(I-1)-$dimensional workload imbalance process, which measures the total amount of work anywhere in the network for stations $1, ..., I-1$ at time t relative to the amount of work in the network at station I at time t. Notice that we have arbitrarily chosen to minimize the long run expected average idleness rate of server 1. Although Z and U are required to be nonanticipating with respect to the $K-$dimensional Brownian motion X, it turns out that they are actually nonanticipating with respect to the $(I-1)$-dimensional Brownian motion B. Constraints (8)-(10) are straightforward , since the cumulative idleness process must be nondecreasing, the customer population size is fixed, and the queue length process must be nonnegative.

3. The Workload Imbalance Polytope and Extremal Classes

For the two-station case, Harrison and Wein (1990) found an optimal solution (Z^*, U^*) to the workload imbalance formulation (5)-(10), and interpreted this solution in terms of the original queueing system in order to find an effective sequencing policy. Unfortunately, we have been unable to find a closed form solution to (5)-(10) when $I > 2$. Instead, we will be satisfied with gaining a deep enough understanding of the problem so that an effective sequencing policy can be found.

We begin this section by verbally describing problem (5)-(10). Define the $(I-1)-$dimensional workload imbalance process $\hat{W} = (\hat{W}_i)$ by

$$\hat{W}_i(t) = \sum_{k=1}^{K} \hat{M}_{ik} Z_k(t), \quad \text{for } i = 1, ..., I-1, \text{ and } t \geq 0. \tag{11}$$

It is clear from equations (9)-(11) that the workload imbalance process must reside within the *workload imbalance polytope* defined by

$$\{(\hat{w}_1, ..., \hat{w}_{I-1}) : \hat{w}_i = \sum_{k=1}^{K} \hat{M}_{ik} z_k, i = 1, ..., I-1; \sum_{k=1}^{K} z_k = 1; z_k \geq 0, k = 1, ..., K\}. \tag{12}$$

This polytope is the convex hull of the K columns of the workload imbalance profile matrix $\hat{M}$, where the k^{th} column of $\hat{M}$ quantifies the workload imbalance of a class k customer.

By equations (7) and (11), it follows that

$$\hat{W}_i(t) = B_i(t) + \rho_I U_i(t) - \rho_i U_I(t), \quad \text{for } i = 1, ..., I-1, \text{ and } t \geq 0. \tag{13}$$

Thus, the workload imbalance formulation can be analyzed in a two-step procedure. The first problem is to find an optimal control U^* (that is nonanticipating with respect to B) to minimize (5) subject to constraints (8) and (13), and subject to the workload imbalance process $\hat{W}$ residing in the workload imbalance polytope defined in (12). The solution U^* to the first problem will lead to an optimal workload imbalance process $\hat{W}^*$ via equation (13) with U^* replacing U, and the second problem is to choose an optimal process Z^* that is nonanticipating with respect to B and satisfies equations (9)-(11), with $\hat{W}^*$ replacing $\hat{W}$ in (11). We will now discuss the two problems in turn.

The first problem is a multidimensional ergodic singular Brownian control problem. The controller observes the $(I-1)-$dimensional Brownian motion B, exerts the nondecreasing controls $U_1, ..., U_I$, and the resulting process is the $(I-1)-$dimensional workload imbalance process given in (13). Notice that the control U_i affects only $\hat{W}_i$, for $i = 1, ..., I-1$, whereas U_I affects the entire process $\hat{W}$. The objective is to exert as little of the controls as possible (recall that we arbitrarily chose to minimize U_1) subject to keeping the controlled process inside the workload imbalance polytope (12). The control problem is described as *singular* because the state of the controlled process can be instantaneously changed by the controller and, as a result, the optimal control process U is continuous but singular (that is, the set of time points at which U increases has measure zero).

When $I = 2$ (see Harrison and Wein 1990), the workload imbalance polytope is a closed interval on the real line, which will be denoted by $[a, b]$, the optimal control processes U_1^* and U_2^* are proportional to the local times at the respective boundaries, and thus the workload imbalance process is a one-dimensional *regulated* or *reflected* Brownian motion (abbreviated hereafter by RBM; see Harrison 1985 for a complete treatment) on the interval $[a, b]$. Since our objective function is to exert the control U as little as possible subject to

keeping $\hat{W}$ in $[a, b]$, it is not surprising that the control U is exerted only when the process $\hat{W}$ reaches the two endpoints of the closed interval.

Unfortunately, closed form solutions to ergodic singular control problems have been restricted to one-dimensional problems (see, for example, Karatzas 1983, Taksar 1985, and Wein 1990a). When $I > 2$, the optimal control U will again only be exerted when the $(I-1)-$dimensional workload imbalance process $\hat{W}$ reaches the boundary of the polytope defined in (12). However, the problem is greatly complicated by the fact that the optimal angle of reflection (exerting different combinations of the components of U yields 2^I possible angles of reflection; see Wein 1990c for details) off the boundary interior must be found. Kushner (1977,1990) has developed a numerical procedure (called the finite difference approximation method in Kushner 1977, and called the Markov chain approximation method in Kushner and Martins 1990) for solving a wide variety of control problems, including multidimensional ergodic singular control problems. By discretizing the state space and time, this technique allows one to approximate our ergodic singular control problem by a finite state Markov chain control problem with a long run average cost criterion, which in turn can be solved numerically using standard techniques. Kushner and Martins (1990) (and references therein) have developed weak convergence methods to prove that, as the discretization of time and space gets finer, the optimally controlled Markov chain (suitably interpolated) converges to the optimally controlled diffusion, and the optimal cost of the controlled Markov chain converges to the optimal cost of the singular control problem. This procedure was used in Wein (1990c) to numerically solve a more difficult constrained ergodic singular control problem arising from a queueing network scheduling problem with controllable inputs. Although we have successfully employed this technique to find numerical solutions to the examples in Section 5, the optimal angles of reflection are not reported here for reasons that will become clear below. However, it is interesting to note that the solution does not appear to be of a simple form, in that the angles of reflection are not constant on each face of the polytope.

We now turn to the second problem in the two step procedure to solve (5)-(10). Given an optimal workload imbalance process $\hat{W}^*$ (via equation (13)) from step one, choose an optimal queue length process Z^* that satisfies constraints (9)-(11), with $\hat{W}^*$ replacing $\hat{W}$ on the left side of equation (11). Let us again begin with the two-station problem considered in Harrison and Wein (1990). In this case, the one-dimensional workload imbalance process $\hat{W}$ is a RBM on the interval $[a, b]$, and $\hat{M}$ is a $K-$dimensional vector, where $\hat{M}_k$ is the workload imbalance for class k. Furthermore, $\min_{\{1 \leq k \leq K\}} \hat{M}_k = a$ and $\max_{\{1 \leq k \leq K\}} \hat{M}_k = b$, and suppose without loss of generality that $\hat{M}_1 = b$ and $\hat{M}_2 = a$, where class 1 is served at station 1 and class 2 is served at station 2. In order to allow the workload imbalance process to evolve in the entire workload imbalance polytope, only the customer classes that correspond to the extreme points of the polytope must have a positive queue length (i.e., $Z_k^*(t) > 0$); the other classes may have a zero queue length for all times t. The customer classes that correspond to the extreme points of the polytope will be called *extremal classes.* In the two-station case, the extreme points of the polytope are a and b, and $\hat{M}_2 = a$ and $\hat{M}_1 = b$, and thus there are exactly two extremal classes, class 1 and class 2. If we force the other $K - 2$ customer classes to have zero queue length (i.e., $Z_k^*(t) = 0$ for $t \geq 0$ and $k = 3, ..., K$), then $Z_1^*(t) = \gamma(t)$ and $Z_2^*(t) = 1 - \gamma(t)$, where $\gamma(t) = (\hat{W}^*(t) - a)/(b - a)$ is the unique solution to equations (9)-(11).

Before we turn to the case where $I > 2$, let us interpret the optimal solution (Z^*, U^*) to the two-station case. The workload imbalance process is a RBM on $[a, b]$, and the server idleness is only incurred when the workload imbalance process equals a or b. Furthermore, only two customer classes, denoted by classes 1 and 2, ever have a positive queue length. Under heavy traffic conditions, it is well-known (see Whitt 1971, Harrison 1973, Reiman 1983, Johnson 1983, Peterson 1985, and Chen and Mandelbaum 1987 for various queueing systems) that if a static priority discipline is used among the customer classes visiting a particular queue, only the lowest priority customer class will have a positive scaled queue length under heavy traffic conditions. The other customer classes will not see the system

in heavy traffic, and thus their queue lengths will be negligible compared to the bottom priority class. Therefore, the solution is interpreted to mean that customers of class 1 (respectively, class 2) are served at station 1 (respectively, station 2) only when there are no other customers present there. Although some ambiguity remains in specifying the entire sequencing policy, the value of $\hat{M}_k$ offers a natural index with which to prioritize the remaining classes. In particular, the proposed workload balancing policy is to award higher priority at station 1 (respectively, station 2) to the classes with the smaller (respectively, larger) values of the index $\hat{M}_k$.

Returning to the case where $I > 2$, in general there are more extremal classes than stations. Thus, unlike the two-station case, there is not a unique combination of the extremal queue lengths $Z_k^*(t)$ that is consistent (in the sense of equation (11)) with the workload imbalance process when it is in the interior of the workload imbalance polytope. Therefore, although the extremal classes can be easily identified, there appears to be many possible solutions Z^* that will allow the workload imbalance process to evolve in the entire workload imbalance polytope. Moreover, since there are more extremal classes than stations, it appears that a dynamic sequencing policy is required, rather than a static policy, as in the two-station case.

To summarize this section, problem (5)-(10) has been decomposed into two problems. The first problem is a multidimensional ergodic singular control problem that does not appear to have a closed form solution. However, it is clear that the the controller exerts the cumulative idleness process U^* only when the workload imbalance process $\hat{W}^*$ reaches the boundary of the workload imbalance polytope. Also, an approximate numerical solution that specifies the optimal angles of reflection off the polytope boundary can be obtained using the Markov chain approximation technique described in Kushner and Martins (1990). The second problem involves finding an optimal queue length process Z^* that is consistent with the optimal workload imbalance process $\hat{W}^*$ derived from the first problem. Although there is not a unique solution to this problem, the extremal classes, which are the only

classes that receive lowest priority at their respective stations, are easily identified.

Because of the nonuniqueness of the solution to the second problem, much ambiguity remains in interpreting the solution to (5)-(10) in terms of the queueing system in order to obtain an effective dynamic sequencing policy. Moreover, it is not clear to us how to use the optimal angles of reflection to identify an effective sequencing policy. Thus, in the remainder of this paper, we will focus on static sequencing policies, and will only briefly discuss possible dynamic policies.

4. Static Sequencing Policies

A static sequencing policy uses a fixed priority ranking of the different customer classes at each server in the network. Perhaps the two most commonly studied static policies are the SEPT and SERPT rules. Under a static policy, only one class will have lowest priority at each server, and hence only I customer classes will have a nonzero queue length in the approximating Brownian model. Thus, the workload imbalance process $\hat{W}$ will reside inside the $(I-1)-$dimensional simplex defined by the I columns of the workload imbalance profile matrix $\hat{M}$ corresponding to the lowest priority classes. This simplex will be contained within the workload imbalance polytope defined in (12).

For any arbitrary static policy, suppose class i is awarded lowest priority at station i, for $i = 1, ..., I$, and thus classes $I+1, ..., K$ are not bottom priority classes. Then for any value $\hat{W}(t)$ of the workload imbalance process in the $(I-1)-$dimensional simplex, there exists a unique nonnegative solution $Z^*(t)$ to the system of equations

$$\hat{W}(t) = \sum_{k=1}^{I} \hat{M}_{ik} Z_k(t), \tag{14}$$

$$\sum_{k=1}^{I} Z_k(t) = 1. \tag{15}$$

Moreover, since idleness would only be incurred at each station when there are no customers present there, in the idealized Brownian model, the control $U_i^*(t)$ is only exerted at times

t when $Z_i^*(t) = 0$. Thus, by equations (13)-(15), the workload imbalance process would behave as a RBM on the simplex generated by the I lowest priority classes. Also, the angles of reflection off each face would be constant; see Chen (1987) for a definition of RBM on a simplex. Readers are referred to Figure 2 of the next section, where two-dimensional simplices are shown for the SEPT and SERPT policies for a specific three-station example.

Recall that the primary performance measure for closed queueing networks is the mean throughput rate, which can be calculated from the mean idleness rates at the various stations. There are several numerical techniques (Harrison, Landau, and Shepp 1981 and Trefethen and Williams 1983 use conformal mapping for the two-dimensional case, and the Markov chain approximation method of Kushner and Martins 1990 can be used for the general case) available for determining the steady state distribution of a RBM on a simplex and the mean rate of pushing off the boundaries, and the latter measure leads directly to an estimate of the mean idleness rate. Thus, we can approximately analyze the performance of any arbitrary static policy, such as SEPT and SERPT. However, these techniques require a substantial effort, perhaps more than many analysts would be willing to undertake in order to just compare different static policies.

As an alternative, we propose a very simple measure to crudely compare various static policies. To motivate our measure, consider the Brownian model of the perfectly balanced two-station closed network. In this case, the drift of the underlying one-dimensional Brownian motion B is zero, and the steady state distribution of the RBM is uniformly distributed over the simplex, which in this case is the closed interval $[a, b]$. Moreover, the average idleness rate (or the average pushing off the two interval endpoints) is the same for each station, and is inversely proportional to $b - a$, the length of the interval (see Harrison and Wein 1990 for a closed form expression).

Now consider the general multistation case. If the RBM was uniformly distributed over the simplex, then a relative measure of the average idleness rate (or pushing off the boundaries) is the surface of the simplex divided by its volume. For example, in a three-

station network, this measure is the perimeter of a triangle divided by its area. This ratio is easy to compute in general, since the volume and the surface of each face can be computed from a determinant. Although our relative measure is correct for the perfectly balanced two-station network, it is a very crude estimate for a multidimensional RBM, since the steady state distribution is not uniform, and the drift, covariance, and angles of reflection of the RBM are being ignored. However, the goal is to develop a very simple measure that hopefully captures the first-order effect that one would observe from a visual inspection of the simplices. Moreover, this approach to performance analysis also extends to a possibly optimal policy, since the numerical solution (via the Markov chain approximation method) to the singular control problem yields the average idleness rate, and a crude estimate of the average idleness rate is just the surface of the workload imbalance polytope divided by its volume. Although we have been unable to identify an optimal policy, one could use this technique to approximately compare the performance of an optimal policy to an arbitrary static policy.

Now that the performance of arbitrary static policies has been discussed, we are now ready to propose an effective static policy. The first step is to find the class from each of the I stations so that the simplex generated by these classes (via the columns of the workload imbalance profile matrix $\hat{M}$) has the minimal ratio of surface-to-volume. For ease of presentation, let us denote these classes by $1, ..., I$, where class i is served at station i. By the above discussion, it is clear that our crude measure of performance would predict that a sequencing policy awarding lowest priority to class i at station i, for $i = 1, ..., I$, would achieve minimal mean idleness, and hence maximal mean throughput, among the class of static policies.

A simple extension to this idea will be used to prioritize classes $I + 1, ..., K$ at their various stations, and hence to complete the specification of the sequencing policy. In order to prioritize the remaining customers at station i, let us suppose for the moment that class i, the lowest priority class at station i, did not exist. Then for each of the remaining classes

at station i, which are indexed by $n = 1, ..., l_i$, we would compute the surface-to-volume ratio R_n for the simplex generated by class n and the remaining $I - 1$ bottom priority classes. Since the class with the smallest value of R_n would receive lowest priority at station i if class i did not exist, our proposed sequencing policy awards higher priority at station i to the classes with the larger values of R_n.

Although one could obtain a more reliable proposed policy by calculating the mean idleenss rate using the sophisticated numerical techniques described earlier in place of our crude surface-to-volume measure, much more computation would be required. Furthermore, as will be seen in the next section, our crude relative idleness measure appears to be accurate enough to distinguish between the various static policies.

We have been unable to identify a simple dynamic policy that significantly outperforms the static policy described above. One policy that was tested in the simulation experiment of the next section was to serve all extremal classes on a first-come first-served basis at their respective stations, and then to prioritize the non-extremal classes in the same order as they were served in the proposed static policy. The hope was that by allowing all extremal classes to have a positive queue length, the workload imbalance process would be allowed to move throughout the entire workload imbalance polytope, as opposed to only moving throughout the simplex of minimal surface-to-volume ratio. However, this policy did not perform significantly better than the proposed static policy, and thus the simulation results for this policy are not reported here.

We have had several ideas for dynamic policies. One is to employ dynamic reduced costs (as in Wein 1990c) derived from the mathematical program of maximizing the minimum amount of work queued at any given station, subject to constraints (14)-(15) and (10), for any given value of $\hat{W}(t)$. A second policy would, given the value of $\hat{W}(t)$ at time t, derive the simplex of minimal surface-to-volume ratio (with one extreme point per station) containing $\hat{W}(t)$, and would award the lowest priority at time t to the classes corresponding to the extreme points of the simplex. The remaining classes would be prioritized at time

t as in the proposed static policy. However, these two policies were not pursued because they would be extremely tedious to implement in a real time setting. Our goal instead is to find a simple and effective sequencing policy.

5. Examples

In this section, simulation results are reported for three example networks, including two three-station networks and a four-station network. Although we believe this procedure remains effective for any number of stations, few factories have more than three or four bottleneck stations, and hence we did not examine larger networks.

There are two objectives in this simulation study: to assess the effectiveness of the proposed static sequencing policy described in Section 4, and to assess the accuracy of the surface-to-volume ratio in predicting the relative performance of various static policies. To achieve the first goal, five sequencing policies are tested for each example: the proposed static policy (denoted by BROWNIAN in the tables below), the first-come first-served (FCFS) policy, the SEPT rule, the SERPT rule, and the least work next queue (LWNQ) rule. This last rule, which gives dynamic priority at each station to the class whose next station has the least amount of work in it, appears to be a reasonable candidate for a closed network setting, where the sole issue is to avoid server idleness.

Recall that the objective of the scheduling problem is to maximize the mean throughput rate for a fixed population level N. In the simulation results below, the population size N for each policy is set so as to achieve a fixed mean throughput rate, and we will instead record the mean sojourn times. As mentioned earlier, minimization of mean sojourn time is equivalent to maximization of mean throughput rate in a closed network. We compare mean sojourn time at a specified throughput rate because this is how factories are generally run: they choose their customer population level to meet the specified exogenous demand rate, and smaller mean sojourn times imply better performance. For each policy, ten independent runs are made, each consisting of 10,000 customer completions and no

initialization periods.

In order to assess the effectiveness of the surface-to-volume ratios in predicting the relative performance of static policies, the SEPT policy, the SERPT policy, and the proposed static policy are tested at constant population levels, and the mean idleness rates are observed. The relative mean idleness rates (all mean idleness rates are divided by the mean idleness rate of the proposed static policy) are then compared to the relative surface-to-volume ratios (the ratio of each policy is divided by the ratio of the proposed policy).

The first network is populated by three types of customers, denoted by A, B, and C, and the specified mix is to have equal numbers of all three types; thus whenever a customer exits the network, the newly injected customer is of type A, B, or C with probability one-third. Table I describes the deterministic route of each customer type, and gives the mean service time for each stage of service. All service time distributions in this section are assumed to be exponential, although our results hold for any general service time distributions. Since each customer class corresponds to a combination of customer type and stage of completion, the twelve customer classes are designated (and ordered from $k = 1, ..., 12$) by (A1,A2,A3,B1,...,B5,C1,...,C4).

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

Table I. Description of example 1.

From Table I, we find that the 3×12 workload profile matrix M is given by

$$M = \begin{pmatrix} 4 & 4 & 0 & 10 & 2 & 2 & 2 & 0 & 4 & 4 & 4 & 0 \\ 1 & 1 & 1 & 13 & 13 & 7 & 7 & 7 & 4 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 11 & 11 & 2 & 2 \end{pmatrix}, \tag{16}$$

where M_{ik} is the expected remaining processing time at station i for a class k customer until that customer exits the network. Since $q = (\frac{1}{3}\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0)^T$, we have $v_1 = v_2 = v_3 = 6$, implying $\rho_1 = \rho_2 = \rho_3 = 1$. Thus, the 2×12 workload imbalance profile matrix $\hat{M}$ is given by

$$\hat{M} = \begin{pmatrix} -2 & 4 & 0 & 9 & 1 & 1 & 2 & 0 & -7 & -7 & 2 & -2 \\ -5 & 1 & 1 & 12 & 12 & 6 & 7 & 7 & -7 & -11 & -2 & -2 \end{pmatrix}. \tag{17}$$

The twelve points $(\hat{M}_{1k}, \hat{M}_{2k})$ are plotted in Figure 1, where the workload imbalance polytope, which is the convex hull of these points, is also displayed. Thus six of the twelve classes are extremal classes, and the number beside each extremal point in Figure 1 is the station that serves the corresponding extremal class. Recall that the static priority policy finds the simplex (containing exactly one point from each station) of minimal surface-to-volume ratio, and awards lowest priority to these three classes at their respective station. Readers can easily see from Figure 1 that the three lowest priority classes are the two highest points (i.e., with maximum $\hat{W}_2(t) = W_2(t) - W_3(t)$ value) and the lowest point, which correspond to class B1 at station 1, class B2 at station 2, and class C2 at station 3. A complete specification of the three static policies is exhibited in Table II. Figure 2 shows the simplices for the SEPT and SERPT policies. A visual inspection reveals that we would expect the SEPT policy to outperform the SERPT policy, since it has a significantly larger simplex.

Simulation results for this example are reported in Tables III and IV. In Table I, the population size, mean sojourn time, and mean throughput rate, along with appropriate 95% confidence intervals, are reported for each of the five sequencing policies, where the throughput rate of .149 customers per unit time corresponds to a server utilization of 89.4%. It can be seen that the proposed static policy easily outperforms the other four policies,

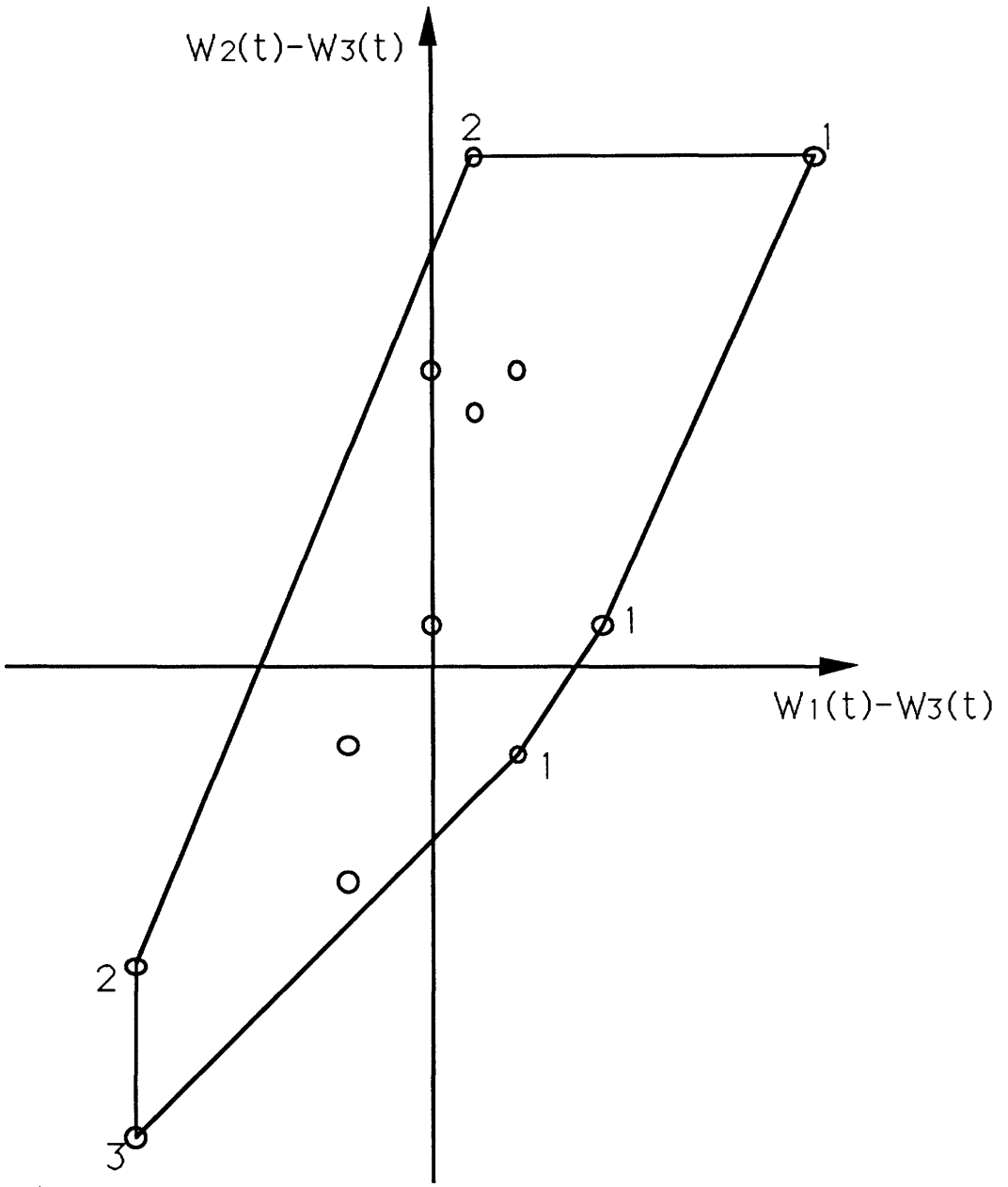

Figure 1. The workload imbalance polytope for example 1.

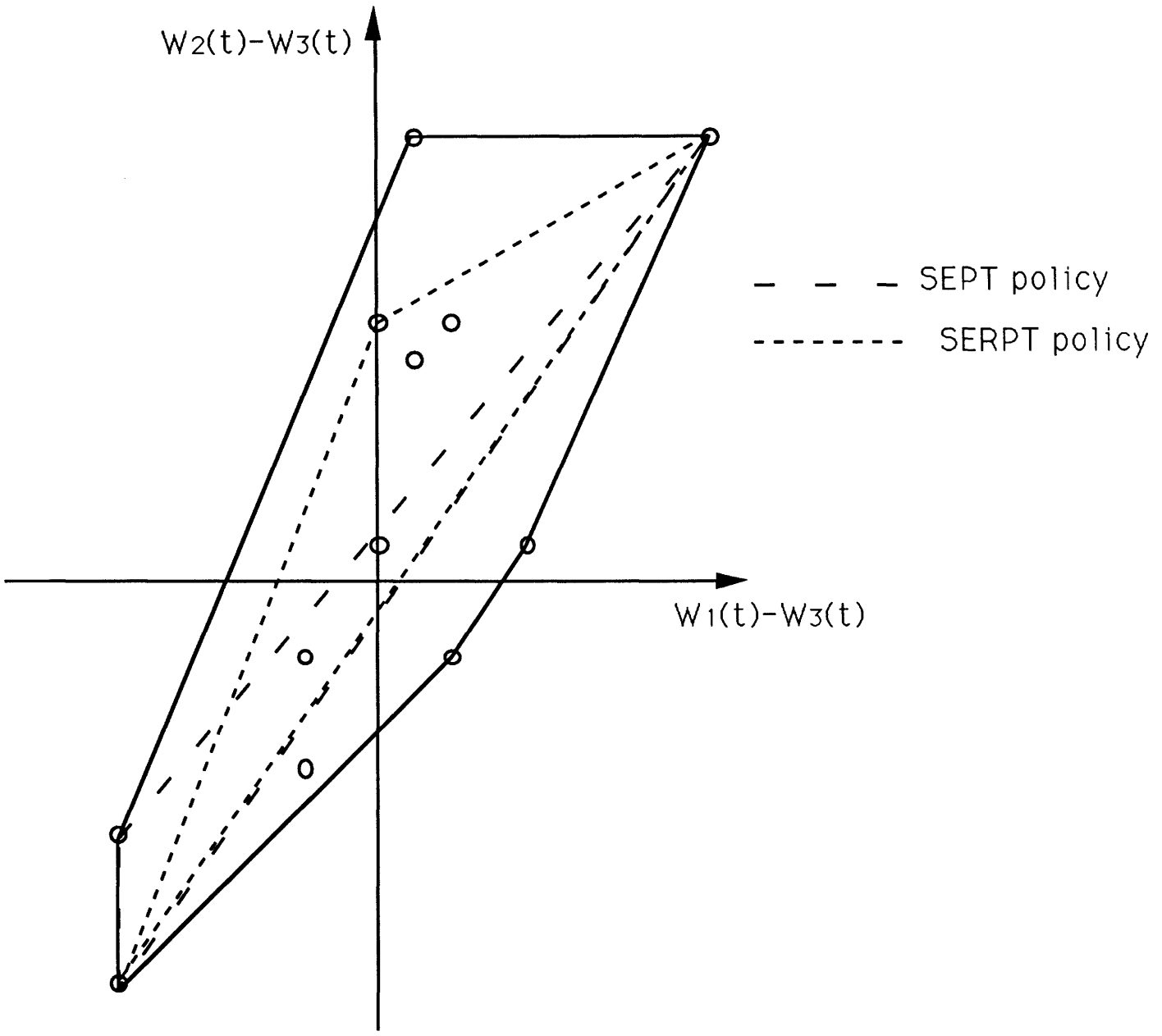

Figure 2. The workload imbalance simplices for SEPT and SERPT.

offering a 43.8% reduction in mean sojourn time versus FCFS. Notice that the LWNQ policy does not offer much improvement over FCFS and, as expected, SEPT outperforms SERPT.

POLICY	STATION 1	STATION 2	STATION 3
BROWNIAN	B4 C3 A2 B1	A3 C1 B5 B2	B3 C4 A1 C2
SEPT	B4 (A2,C3) B1	A3 C1 B2 B5	B3 C4 A1 C2
SERPT	A2 C3 B4 B1	A3 B5 B2 C1	C4 B3 A1 C2

Table II. Static sequencing policies for example 1.

SEQUENCING POLICY	POPULATION SIZE	MEAN SOJOURN TIME	MEAN THROUGHPUT
BROWNIAN	14	93.8 ($\pm$0.57)	.149 ($\pm$.0008)
SEPT	20	134 ($\pm$0.66)	.149 ($\pm$.0007)
LWNQ	24	161 ($\pm$1.05)	.149 ($\pm$.0010)
FCFS	25	167 ($\pm$1.05)	.149 ($\pm$.0010)
SERPT	30	201 ($\pm$1.20)	.149 ($\pm$.0009)

Table III. Comparison of mean sojourn times for example 1.

In Table IV, the three static policies are compared at three different population levels, and the observed actual idleness rates, normalized so that the idleness rate of the proposed policy is one, are compared to the normalized versions of the estimated idleness rates (via the surface-to-volume ratios). For example, we predict that the SERPT rule will have 2.7 times as much idleness as the BROWNIAN policy when the population size is very large. When the population size is 45, the SERPT rule actually incurs 2.6 times as much idleness as the BROWNIAN policy, and thus the surface-to-volume ratio is quite accurate

in this case. Although the ratio is not an accurate predictor in the SEPT case, the measure correctly predicts the relative performance of the three policies.

STATIC SEQUENCING POLICY	POPULATION SIZE	NORMALIZED IDLENESS RATE	NORMALIZED SURFACE-TO-VOLUME RATIO
BROWNIAN	15	1.00	1.00
SEPT	15	1.39	1.38
SERPT	15	1.68	2.70
BROWNIAN	30	1.00	1.00
SEPT	30	1.60	1.38
SERPT	30	2.30	2.70
BROWNIAN	45	1.00	1.00
SEPT	45	1.77	1.38
SERPT	45	2.60	2.70

Table IV. Actual and predicted normalized idleness rates for example 1.

Before turning to example 2, we want to mention that the workload imbalance polytope can be helpful in developing a fast heuristic solution to a related scheduling problem considered in Wein (1990c). This study develops a customer release and priority sequencing policy to minimize mean sojourn time subject to a minimum mean throughput rate constraint. The resulting constrained singular ergodic Brownian control problem is to find a region in R^{I-1} in which to reflect the workload imbalance process. When there is perfect balance between the stations in the two-station case, it turns out that the region derived in the controllable inputs problem is homothetic (that is, of similar shape) to the workload imbalance polytope of the corresponding closed network problem. Moreover, this relationship appears to roughly hold in the multistation case; readers may compare

the similarity in shapes of the workload imbalance polytope in Figure 1 with the optimal reflecting boundary in Figure 5 of Wein (1990c), which also considered the network described in Table I. This is significant because the numerical solution that derives the optimal reflecting boundary is extremely difficult to obtain for networks with more than three bottleneck stations, whereas the workload imbalance polytope is relatively easy to obtain for any size network.

Example 2 is also a three-station network visited by three customer types. The customer routes and mean service times are given in Table V, and the mix of customer types is again $(1/3, 1/3, 1/3)$. Readers may verify that $\rho_1 = \rho_2 = \rho_3 = 1$, and the workload imbalance profile matrix is given by

$$\hat{M} = \begin{pmatrix} -1 & -2 & 4 & 4 & -1 & -4 & -4 & 2 & -3 & -3 \\ -1 & -1 & 5 & 0 & 2 & 2 & -4 & -1 & -1 & -3 \end{pmatrix}. \tag{18}$$

Although we do not exhibit the simplices for the static policies here, a visual inspection of these simplices suggests that the BROWNIAN policy should outperform the SEPT policy, which in turn should outperform the SERPT policy.

CUSTOMER TYPE	ROUTE	MEAN SERVICE TIMES
A	$1 \rightarrow 3 \rightarrow 2 \rightarrow 1$	1.0 6.0 5.0 4.0
B	$1 \rightarrow 2 \rightarrow 3$	3.0 6.0 4.0
C	$1 \rightarrow 2 \rightarrow 3$	5.0 2.0 3.0

Table V. Description of example 2.

Simulation results for Example 2 are found in Tables VI and VII. The mean throughput rate of .210 in Table VI corresponds to a mean server utilization of 91.1%. Once again, the BROWNIAN policy outperforms the other four policies, and offers a 32.2% reduction in mean sojourn time versus FCFS. The LWNQ policy did not perform as well as FCFS and,

as predicted, SEPT outperformed SERPT. The normalized ratios clearly overestimate the normalized idleness rates in Table VII. However, the relative values of the three normalized idleness rates were predicted reasonably accurately when $N = 45$, since (2.46-1.00)/(6.10-1.00)=.286, and (1.67-1.00)/(2.93-1.00)=.347.

SEQUENCING POLICY	POPULATION SIZE	MEAN SOJOURN TIME	MEAN THROUGHPUT
BROWNIAN	17	80.7 (±0.37)	.210 (±.0009)
SEPT	22	105 (±0.62)	.210 (±.0013)
FCFS	25	119 (±0.55)	.210 (±.0010)
LWNQ	29	138 (±0.75)	.210 (±.0011)
SERPT	45	213 (±1.44)	.210 (±.0014)

Table VI. Comparison of mean sojourn times for example 2.

STATIC SEQUENCING POLICY	POPULATION SIZE	NORMALIZED IDLENESS RATE	NORMALIZED SURFACE-TO-VOLUME RATIO
BROWNIAN	15	1.00	1.00
SEPT	15	1.21	2.46
SERPT	15	1.74	6.10
BROWNIAN	30	1.00	1.00
SEPT	30	1.27	2.46
SERPT	30	2.40	6.10
BROWNIAN	45	1.00	1.00
SEPT	45	1.67	2.46
SERPT	45	2.93	6.10

Table VII. Actual and predicted normalized idleness rates for example 2.

Our last example is the four-station network described in Table VIII. There are four customer types and a total of twenty customer classes. A newly injected customer is of each type with probability .25, and thus the network is again perfectly balanced ($\rho_i = 1$ for $i = 1, ..., 4$).

CUSTOMER TYPE	ROUTE	MEAN SERVICE TIMES
A	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2.0 4.0 3.0 7.0
B	$4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1$	3.0 5.0 2.0 4.0 1.0 6.0
C	$2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 3 \rightarrow 2$	2.0 8.0 2.0 9.0 5.0 6.0
D	$2 \rightarrow 4 \rightarrow 1 \rightarrow 3$	2.0 1.0 2.0 6.0

Table VIII. Description of example 3.

The simulation results for example 3 are displayed in Tables IX and X. As can be seen from Table X, the normalized surface-to-volume ratio for the SEPT policy is only 1.28 in this case, and so we would predict that the difference in performance between the BROWNIAN and SEPT policies would be less in this example than in the previous two examples. This prediction is verified in Table IX, where the desired throughput rate is .165, which corresponds to a server utlization rate of only 82.4%. Since the SEPT policy was unable to achieve this rate exactly, we have included two rows in Table IX for this policy, where each row uses a different population size.

It is interesting to note that in Tables IV, VII, and X, the normalized idleness rates of SEPT and SERPT increase as the population size increases, and thus the BROWNIAN policy's relative performance is better at higher population levels. This may be due in part because the policy is derived under balanced heavy loading conditions, and in part because, as in open networks, the improvements from scheduling may increase as network congestion increases. Thus, the relatively low server utilization in Table IX may also

contribute to the similarity in performance between the BROWNIAN and SEPT policies.

Once again, the BROWNIAN policy offers a significant reduction (38.0%) in mean sojourn time versus FCFS. There is a very wide range of performance among the policies, with the SERPT policy possessing a mean sojourn time that is 7.6 times larger than that of the BROWNIAN policy. In this example, the normalized surface-to-volume ratios underestimated the normalized idleness rates at $N = 45$, although the relative values of the three normalized idleness rates were accurately predicted, since (1.28-1.00)/(4.93-1.00)=.071, and (1.44-1.00)/(7.00-1.00)=.073.

SEQUENCING POLICY	POPULATION SIZE	MEAN SOJOURN TIME	MEAN THROUGHPUT
BROWNIAN	13	78.8 (±0.37)	.165 (±.0010)
SEPT	13	79.4 (±0.50)	.164 (±.0010)
SEPT	14	84.4 (±0.43)	.166 (±.0008)
FCFS	21	127 (±0.73)	.165 (±.0014)
LWNQ	55	332 (±2.42)	.165 (±.0012)
SERPT	100	601 (±6.34)	.165 (±.0017)

Table IX. Comparison of mean sojourn times for example 3.

STATIC SEQUENCING POLICY	POPULATION SIZE	NORMALIZED IDLENESS RATE	NORMALIZED SURFACE-TO-VOLUME RATIO
BROWNIAN	20	1.00	1.00
SEPT	20	1.09	1.28
SERPT	20	2.37	4.93
BROWNIAN	40	1.00	1.00
SEPT	40	1.29	1.28
SERPT	40	4.58	4.93
BROWNIAN	60	1.00	1.00
SEPT	60	1.44	1.28
SERPT	60	7.00	4.93

Table X. Actual and predicted normalized idleness rates for example 3.

We should note that although the SEPT policy outperformed the SERPT policy in all three examples, counterexamples to this phenomenon can be easily constructed. Readers are referred to the two-station closed network example in Harrison and Wein (1990), where the SEPT policy is easily outperformed by SERPT. However, the Brownian analysis does explain why the SERPT policy will often perform poorly in a closed network under balanced heavy loading conditions. The lowest priority class at each station under SERPT is the class with the maximum value of $\sum_{i=1}^{I} M_{ik}$, and these classes usually correspond to the early stages on the customers' routes. Since Mq is proportional to the vector ρ of traffic intensities (whose components are close to each other in value by the balanced heavy loading conditions), these classes will not often be extremal classes of the workload imbalance polytope, unless there are significant differences in workload imbalance across

entering customer types.

In summary, we have analyzed a Brownian approximation to the scheduling problem of maximizing the mean throughput rate of a general multistation, multiclass closed queueing network. The insights gained from this analysis have led to an identification of an effective static policy, and to a crude but robust procedure for predicting the performance of an arbitrary static sequencing policy. We believe the most interesting aspect of this study is the dramatic impact that different static policies can have on system performance.

Acknowledgements

This research is partially supported by a grant from the Leaders for Manufacturing Program at MIT.

REFERENCES

Baskett, F., K. M. Chandy, R. R. Muntz, and F. G. Palacios. 1975. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. *J. Assoc. Comput. Mach.* **22**, 248-260.

Chen, H. 1987, Stochastic Flow Networks: Bottleneck Analysis, Fluid Approximations, and Diffusion Limits. Unpublished Ph.D. thesis, Dept. of Engineering-Economic Systems, Stanford U., Stanford, CA.

Chen, H. and A. Mandelbaum. 1987. Stochastic Discrete Flow Networks: Diffusion Approximations and Bottlenecks. Submitted to *Annals of Probability*.

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

Harrison, J. M. 1985. *Brownian Motion and Stochastic Flow Systems.* John Wiley and Sons, New York.

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.

Harrison, J. M., H. J. Landau, and L. A. Shepp. 1985. The Stationary Distribution of Reflected Brownian Motion in a Planar Region. *Annals of Probability* **13**, 744-757.

Harrison, J. M. and L. M. Wein. 1989. Scheduling Networks of Queues: Heavy Traffic Analysis of a Simple Open Network. *Queueing Systems* **5**, 265-280.

Harrison, J. M. and L. M. Wein. 1990. Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network. To appear in *Operations Research.*

Johnson, D. P. 1983. Diffusion Approximations for Optimal Filtering of Jump Processes and for Queueing Networks. Unpublished Ph.D. thesis, Dept. of Mathe-

matics, Univ. of Wisconsin, Madison.

Karatzas, I. 1983. A Class of Singular Stochastic Control Problems. *Adv. Appl. Prob.* **15**, 225-254.

Kelly, F. P. 1979. *Reversibility and Stochastic Networks*, John Wiley and Sons, New York.

Klimov, G. P. 1974. Time Sharing Service Systems I. *Th. Prob. Appl.* 19,532-551.

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

Kushner, H. J. 1990. Numerical Methods for Stochastic Control Problems in Continuous Time. To appear in *SIAM J. Control and Optimization.*

Kushner, H. J. and F. L. Martins. 1990. Numerical Methods for Stochastic Singularly Controlled Problems. Technical Report, Div. Applied Math., Brown U., Providence, R. I.

Little, J. D. C. 1961. A Proof of the Queueing Formula $L = \lambda W$. *Operations Research* **9**, 383-387.

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

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

Taksar, M. I. 1985. Average Optimal Singular Control and a Related Stopping Problem. *Mathematics of Operations Research* **10**, 63-81.

Trefethen, L. N. and R. J. Williams. 1986. Conformal Mapping Solution of Laplace's Equation on a Polygon with Oblique Derivative Boundary Conditions. *J. Comp. Appl. Math.* **14**, 227-249.

Wein, L. M. 1990a. Optimal Control of a Two-Station Brownian Network. *Mathematics of Operations Research* **15**, 215-242.

Wein, L. M. 1990b. Scheduling Networks of Queues: Heavy Traffic Analysis of a

Two-Station Network With Controllable Inputs. To appear in *Operations Research*.

Wein, L. M. 1990c. Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Network With Controllable Inputs. Submitted to *Operations Research*.

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