Dynamic Scheduling in Single-Server Multiclass Service Systems with Unit Buffers

Sid Browne
Graduate School of Business, Columbia University, New York, New York 10027

Uri Yechiali
Department of Statistics, Tel Aviv University, Tel Aviv 69978, Israel

The problem of optimal dynamic sequencing for a single-server multiclass service system with only unit storage (buffer) space at *each* queue is considered. The model is applicable to many computer operating and telecommunicating systems (e.g., polling systems). Index policies to minimize costs for the special case of symmetric arrival rates are derived. Simulations suggest that using these indices provides a substantial improvement over cyclic schedules.

1. INTRODUCTION

Consider a system composed of N queues or channels serviced by a single server. Arrivals to channel i are governed by a homogeneous Poisson process with random, channel-specific, general service requirement distributed as V_i, $i = 1, \ldots, N$. Each channel can store at most one request at a time and all arrivals to a channel that find the "buffer" full (occupied) are lost to the system forever. An occupied channel (queue) reopens only upon the completion of the occupier's service request. Thus, we may equivalently consider a system that takes an exponentially distributed amount of time to generate a new service request at each queue after the service to the previous job has been completed.

This type of system describes the workings of many computer and communications systems. Specifically, for a server that moves in a fixed cyclical fashion around the queues [i.e., following the fixed template $(\ldots j, j+1, \ldots, N-1, N, 1, 2, \ldots)$], it has been analyzed as a polling system. The literature on this topic is immense and rather than refer to it directly, we call the reader's attention to the excellent monograph of Takagi [11] (which is updated in [12]) for a thorough analysis and bibliography on the model, as well as a wealth of applications.

A feature that complicates the analysis of these systems is the nonnegligible "switching time" the server incurs in moving from one channel to another. As such, almost all analyses have concentrated on cases where the server follows the fixed cyclic route, stopping only to service the occupied channels. These

Naval Research Logistics, Vol. 38, pp. 383–396 (1991)
 CCC 0028-1441/91/030383-14$04.00

studies [11] have concentrated on evaluating various performance criteria (e.g., waiting times) for mostly special cases, while probabilistic results for the fully asymmetric systems have only recently been derived (see Takagi [12] and Takine, Takehashi, and Hasegawa [13]). The optimal dynamic scheduling of this model appears to be an open problem, and only recently (see Browne [2] and Browne and Yechiali [3]) has the problem been considered for systems with infinite buffers.

In this article we examine the problem of what route the server should follow if he is allowed to choose his path, or schedule, across the channels at suitably defined decision epochs to optimize a measure of system performance. We assume the server has full system-state information; that is, he knows which channels are occupied at every decision epoch. Admittedly this assumption might nullify the application to local area networks (see Takagi [11]), unless we assume further that token passing delays (switching times) are insignificant relative to packet transmission (service) times. However, we will derive index policies for the semisymmetric case (identical arrival rates with arbitrary service times) that can be used to prioritize the channels and could thus be used in conjunction with general polling tables, as discussed in Baker and Rubin [1].

The remainder of the article is organized as follows: In Section 2 we model the fully asymmetric system as a semi-Markov decision process and find that analytic problems preclude a simple analytical solution. We are therefore led to consider optimization over a Hamiltonian tour of the semisymmetric system in Section 3, which leads to an index rule. We generalize the index to account for switching times in Section 4. In Section 5 we consider the dynamic implementation of the index rule, and evaluate its effectiveness by reporting results of a set of simulations. We conclude with a comparison with related results for infinite buffer systems.

2. FORMULATION

In the general single-buffer multiclass (parallel) queuing system, the ith queue or channel is characterized by an independent Poisson arrival stream of intensity λ_i, an independent service requirement distributed as V_i with distribution function $G_i(\cdot)$, and a switching time. As our interest lies in determining optimal server paths, we assume that all channels i and j are connected by a switching time S_{ij} that decomposes additively into $S_{ij} = \theta_i + \tau_j$, where θ_i denotes the time the server needs to switch out of channel i and τ_j denotes the switch-in or setup time for channel j.

At most one outstanding message or service requirement can be stored at each channel, with those messages arriving to find the buffer full being lost to the system forever. The buffer frees up only at the termination of the occupiers' service.

The following cost structure is imposed on the system: a holding cost at rate \$$h_i$ per unit time a type i is held in queue, and a penalty cost consisting of a payment of \$$b_i$ per type i lost to the system, $i = 1, \ldots N$. The penalty cost could denote the entrance fee to a secondary transmission network that accepts the overflows of the primary system.

To ease exposition and illustrate some basic ideas, we will first analyze the system with zero switching times ($\theta_i = \tau_i = 0, \forall_i$). Let $c_i(s, t)$ denote the total cost incurred at channel i in the (time) interval $(s, t]$ *without* channel i having been serviced in said interval. Let $Q_i(s)$ denote the state of channel i at the initial time s; that is,

$$Q_i(s) = \begin{cases} 1 \text{ if buffer } i \text{ is occupied at } s, \\ 0 \text{ if buffer } i \text{ is vacant at } s. \end{cases}$$

As

$$E(c_i(s, t)|Q_i(s) = 1) = [h_i + \lambda_i b_i](t - s) \tag{1}$$

and

$$E(c_i(s, t)|Q_i(s) = 0) = \int_s^t \lambda_i e^{-\lambda_i(x-s)}[h_i + \lambda_i b_i](t - x)\, dx$$

$$= [h_i + \lambda_i b_i]\left((t - s) - \frac{1 - e^{-\lambda_i(t-s)}}{\lambda_i}\right), \tag{2}$$

we may write

$$E(c_i(s, t)|Q_i(s)) = [h_i + \lambda_i b_i]\left((t - s) - \frac{1 - e^{-\lambda_i(t-s)}}{\lambda_i}(1 - Q_i(s))\right). \tag{3}$$

Consider now the general problem where the server is faced at the decision epoch s with the state vector $\mathbf{Q}(s) = (Q_1(s), \ldots, Q_N(s))$. We restrict our attention to the class of nonidling as well as nonpreemptive policies, so the only available action the server may choose is to serve a channel j for which $Q_j(s) = 1$, and then to stay at j until that service is completed.

To ease notation, let $\mathbf{Q}(s) \equiv \mathbf{Y} \equiv (Y_1, \ldots, Y_N)$. Under action j, the next decision epoch will occur at time $s + V_j$, with resulting state $\mathbf{Q}(s + V_j) = \mathbf{Y}' = (Y_1', \ldots, Y_N')$, where $Y_j' = 0$.

Conditional upon $\mathbf{Y}$, the expected cost to channel i under action j is [directly from Eq. (3)]

$$(h_i + \lambda_i b_i)\left[E(V_j) - \frac{1 - \tilde{V}_j(\lambda_i)}{\lambda_i}(1 - Y_i)\right], \tag{4}$$

where $\tilde{V}_j(\alpha)$ denotes $E(e^{-\alpha V_j})$, the Laplace–Stieltjes transform of V_j at α.

The transition probability for channel i under action j, conditional on V_j, can be written as

$$E(P_{Y_i, Y_i'}(j)|V_j) = [(1 - Y_i)e^{-\lambda_i V_j(1 - Y_i')}(1 - e^{-\lambda_i V_j})^{Y_i'} + Y_i Y_i'].$$

The state transition probabilities are then

$$P_{Y,Y'}(j) = E\left(\prod_{i \neq j} E(P_{Y_i,Y_i'}(j)|V_j)\right). \tag{5}$$

We have now a standard semi-Markov decision process with average cost optimality equation (see, e.g., Ross [10, p. 162]).

$$H(\mathbf{Y}) = \min_{\{j: Y_j = 1\}} \left\{ \sum_{i=1}^{N} (h_i + \lambda_i b_i) \left[E(V_j) - \frac{1 - \tilde{V}_j(\lambda_i)}{\lambda_i} (1 - Y_i) \right] + \sum_{\mathbf{Y}'} P_{Y,Y'}(j) H(\mathbf{Y}') - gE(V_j) \right\}, \tag{6}$$

where if a bounded function H and a constant g exist to satisfy (6), then a stationary policy exists such that it prescribes an action to minimize the RHS of (6) for each $\mathbf{Y}$.

While one may now use brute-force methods on Eq. (6) to solve for full optimality, the forms of the transition probabilities of Eq. (5) are in general too complex to yield simple-form solutions. In fact Katehakis and Derman [8] as well as Nash and Weber [9] considered a simpler, somewhat related, problem for only the *exponential* distribution and found that the (in that case *Markov*) dynamic programming approach did not yield to solution. In our terminology, they considered the case of preemptive service with a constant cost rate whenever *any* customer was in the system (regardless of how many). Using novel methodologies they were able to solve the problem only for the *exponential* service-time case.

These problems lead us to search for a good heuristic with simple structure that is useful for any size N (such as an index policy) with which to operate the system.

3. OPTIMAL TOURS: THE SEMISYMMETRIC CASE

One family of heuristics can be described as the "pseudocyclic" class (see [2, 3]), which restricts the server to continuously complete cycles or (Hamiltonian) tours of the index set $I = \{1, \ldots, N\}$, but allows him to choose the order in which he visits the channels within each tour. The dynamic implementation of this type of policy will be discussed more fully in the next section, but recognize that an element of fairness across channels is implicit in that no channel need wait longer than one tour to be served. We will first consider the problem of how to optimally perform *one* tour. Specifically, what *policy* achieves the tour of minimal *total* cost? Assume for now that this tour starts at time 0 with all buffers full; that is $\mathbf{Q}(0) \equiv \mathbf{1} = (1, \ldots, 1)$. It is clear by the definition of a (Hamiltonian) tour and our restriction to nonpreemptive service that this (optimal) policy is merely (at least) one of the $N!$ possible *permutations* of I. This is an optimal scheduling problem with a very complicated cost function, as the probability of new arrivals during a tour causes, in general, a high degree of nonlinearity.

Fortunately, the problem becomes tractable under the simplifying assumption that the system is *semisymmetric* in that arrival rates are equal across channels; that is, $\lambda_i = \lambda$, $\forall i$, which we now examine.

It is clear that the *time* to perform this cycle—or tour—say T, is invariant with respect to policy as

$$T = \sum_{i=1}^{N} V_i. \tag{7}$$

Consider now the permutation or schedule $\pi_0 = (1, 2, \ldots, N)$, and note that by definition of π_0 channel i will be occupied until time $\Sigma_{j=1}^{i} V_j$, whereupon its buffer frees up at its service completion, so that $Q_i(\Sigma_{j=1}^{i} V_j) = 0$. We can therefore use Eqs. (1) and (2) directly (with $\lambda_i = \lambda$, $\forall i$) to evaluate the expected total cost incurred by channel i under π_0 as

$$E(c_i(0, T)|\mathbf{Q}(0) = \mathbf{1}, \pi_0)$$

$$= E\left\{(h_i + \lambda b_i) \sum_{j=1}^{i} V_j + (h_i + \lambda b_i)\left[(T - \sum_{j=1}^{i} V_j) - \frac{1 - e^{-\lambda(T - \Sigma_{j=1}^{i} V_j)}}{\lambda}\right]\right\}$$

$$= (h_i + \lambda b_i)(E(T) - \frac{1}{\lambda}) + \left(\frac{h_i + \lambda b_i}{\lambda}\right) E(e^{-\lambda \Sigma_{j=i+1}^{N} V_j})$$

$$= (h_i + \lambda b_i)\left[E(T) - \frac{1}{\lambda}\right] + \left(\frac{h_i + \lambda b_i}{\lambda}\right) \prod_{j=i+1}^{N} \tilde{V}_j(\lambda). \tag{8}$$

The total expected cost incurred by the system following tour π_0 is therefore

$$E(C(0, T)|\mathbf{Q}(0) = \mathbf{1}, \pi_0)$$

$$= \left(E(T) - \frac{1}{\lambda}\right) \sum_{i=1}^{N} (h_i + \lambda b_i) + \sum_{i=1}^{N} \left(\frac{h_i + \lambda b_i}{\lambda}\right) \prod_{j=i+1}^{N} \tilde{V}_j(\lambda). \tag{9}$$

As only the second term in Eq. (9) is affected by policy; it is that term we need to minimize. Consider therefore the policy $\pi_1 = (1, \ldots, r-1, r+1, r, r+2, \ldots, N)$; that is, simply interchange the rth and $r+1$st terms in π_0. It is then straightforward to show that

$$E(C(0, T)|\mathbf{Q}(0) = \mathbf{1}, \pi_0) - E(C(0, T)|\mathbf{Q}(0) = \mathbf{1}, \pi_1)$$

$$= \left[(h_{r+1} + \lambda b_{r+1})\left(\frac{1 - \tilde{V}_r(\lambda)}{\lambda}\right) - (h_r + \lambda b_r)\left(\frac{1 - \tilde{V}_{r+1}(\lambda)}{\lambda}\right)\right] \prod_{j=r+2}^{N} \tilde{V}_j(\lambda), \tag{10}$$

from which it follows that policy π_1 improves upon π_0 iff

$$\lambda \frac{h_{r+1} + \lambda b_{r+1}}{1 - \tilde{V}_{r+1}(\lambda)} > \lambda \frac{h_r + \lambda b_r}{1 - \tilde{V}_r(\lambda)}. \tag{11}$$

Iteration of the pairwise interchange is sufficient to prove the following theorem.

THEOREM 1: The tour of minimal expected cost is prescribed by the policy π^*, which is ordered by *decreasing* values of the index

$$\lambda \frac{h_i + \lambda b_i}{1 - \tilde{V}_i(\lambda)}. \tag{12}$$

DISCUSSION: Note that as $\lambda \searrow 0$, the above policy reduces directly (as expected) to the classical "weighted shortest expected processing time first" (W.S.E.P.T.) (see, e.g., Conway, Maxwell, and Miller [5]) as

$$\lim_{\lambda \searrow 0} \lambda \frac{h_i + \lambda b_i}{1 - \tilde{V}_i(\lambda)} = \frac{h_i}{E(V_i)}.$$

However, for $\lambda > 0$, while $h_i + \lambda b_i$ can be considered the effective cost rate per unit time for a serviced arrival (with no other costs being incurred), $1 - \tilde{V}_i(\lambda)$ is the probability of at least one arrival *during* a type i service. As such, π^* is sensitive to *all* the moments of V_i [assuming $G(\cdot)$ is determined by all its moments], whereas W.S.E.P.T. depends only on $E(V_i)$.

4. SWITCHING TIMES

To incorporate switching times into the model, assume the server takes an independent random time τ_i to switch in or setup at channel i prior to service, and an independent random time θ_i to switch out of channel i postservice there. We further assume the channel, or buffer, frees up at the termination of a service, that is, if the server enters channel i at t (obviously, it must have been occupied at t), the buffer will reopen at time $t + \tau_i + V_i$, and the server is free of, or completes, channel i at time $t + \tau_i + V_i + \theta_i$. Because the buffer is still occupied during the switch-in period, τ_i can be easily incorporated into the service time and we may equivalently let $W_i = \tau_i + V_i$ denote the new service requirement. Let Z_i denote the completion time of channel i in π_0; that is

$$Z_i = \sum_{j=1}^{i} (W_j + \theta_j).$$

Then, using Eqs. (3) and (8) where now $T \triangleq Z_N$, we may write the total expected cost to channel i following π_0 as

$$
\begin{aligned}
&E(c_i(0, T)|\mathbf{Q}(0) = \mathbf{1}, \pi_0) \\
&= E\Big\{(h_i + \lambda b_i)(Z_{i-1} + W_i) + (h_i + \lambda b_i) \\
&\times \Big[(T - Z_{i-1} - W_i) - \frac{1 - e^{-\lambda(T - Z_{i-1} - W_i)}}{\lambda}\Big]\Big\} \\
&= (h_i + \lambda b_i)\Big[E(T) - \frac{1}{\lambda}\Big] + \Big(\frac{h_i + \lambda b_i}{\lambda}\Big)\ E(e^{-\lambda(\theta_i + \Sigma_{j=i+1}^{N}(W_j + \theta_j))}) \\
&= (h_i + \lambda b_i)\Big(E(T) - \frac{1}{\lambda}\Big) + \Big(\frac{h_i + \lambda b_i}{\lambda}\Big)\ \tilde{\theta}_i(\lambda) \prod_{j=i+1}^{N} \tilde{W}_j(\lambda)\tilde{\theta}_j(\lambda).
\end{aligned}
\tag{13}
$$

A pairwise interchange, as in Eq. (10), suffices to prove the following.

THEOREM 2: When switching times are included, the tour of minimal expected cost is prescribed by π^*, which is ordered in decreasing values of the index

$$\lambda \frac{(h_i + \lambda b_i)\tilde{\theta}_i(\lambda)}{1 - \tilde{W}_i(\lambda)\tilde{\theta}_i(\lambda)}. \tag{14}$$

5. DYNAMIC OPERATING POLICIES

5.1. Heuristics

The fact that we obtained index policies for the problem of optimally scheduling the first tour is a great help in the search for good heuristics to operate the system dynamically, as index policies are in general easy to implement. As an example, consider the following implementation of a pseudocyclic policy where the server must always complete a Hamiltonian tour on I before proceeding to the next tour and *always chooses the tour of minimal cost*. For example, suppose he just completed a tour at time t and observes the state vector $\mathbf{Q}(t)$, with $Q_i(t)$ being 0 or 1 if channel i is, respectively, vacant or occupied at t, and say that K channels (buffers) are occupied. The next tour [which must be determined at t based on $\mathbf{Q}(t)$] with minimal cost is to "serve" first the currently vacant channels taking zero time at zero cost and then serve the currently occupied K channels in decreasing order of the index (14), (12), thus completing a tour. At the termination of this tour (i.e., these K steps), say at time $t + \tau$, he is thereupon faced with the new state $\mathbf{Q}(t + \tau)$ containing, say, J occupied buffers. The server now chooses the next tour of minimal cost, which, in effect, is simply the next J steps ahead, and so on. The server idles only when no customers are in the system. In this manner, the server effectively utilizes a *static* index over *dynamic* (decision) horizons dictated by the current workload in the system. There is an element of fairness built into this policy, as an occupied channel will not have to wait longer than one tour to be served.

Another attractive heuristic results if we utilize the index but drop the pseudocyclic, or Hamiltonian, restriction. Consider the case where each channel completion epoch affords the server a decision epoch; that is, he is always allowed to choose the next channel to be served. The server may use the index results to operate the system under a simple *strict absolute priority* scheme, where the server always chooses to serve the occupied channel with maximal value of the index (14). This policy is easier to implement than a pseudocyclic one as it requires no memory about the previous tours. Note though that this policy results in a different priority scheme in general than the more classical $c\mu$ rule (see [5, 7, 14]). This heuristic will be evaluated via a simulation study in the next section, where for obvious reasons we will refer to it as *greedy*.

5.2. Performance Evaluation—Simulation Study

The evaluation of the relative merits of the policies suggested above appears to necessitate a simulation study, as the associated analytics seems to be intractable at present. For example, while the system does regenerate (e.g., each time the system empties), even the mean "busy period" for the much simpler $M/G/1/N$ system is not available in closed form. Furthermore, it is a direct consequence of the work of Katehakis and Derman [8] and Nash and Weber [9] that for general service times, the system busy period remains *policy dependent* (as well as unknown at present) even for symmetric arrivals, leading to major analytical problems. To get some idea though of the effectiveness of our heuristics we resorted to a simulation study. The design is based loosely on the methodology of Coffman and Gilbert [4], who studied a limiting continuous version of a *fully symmetric* system. Their objective was to study the effectiveness of a "greedy" server (i.e., one who always serves via nearest neighbor) versus one who just polled continuously in a cyclic manner. They concluded after a simulation study that in heavily loaded systems, the cyclic server was superior, while in a system with light loads the greedy server outperformed the cyclic one.

In our study, we chose to compare the second heuristic mentioned in Section 5.1 with a strictly cyclic server; that is, we compared a server following a strict absolute priority scheme based on index (12) with one who always completed the same identical cyclic tours [i.e., $(1, \ldots, N-1, N, 1, 2, \ldots, N, 1, \ldots)$]. As a strict absolute priority scheme is a form of a greedy algorithm, we refer to it so below. We simulated a 10-station system under 100 different load factors. Without loss of generality, we set $\lambda = 1$ for all cases and assumed zero switching times. As such, we define the *load factor* of each run to be $\Sigma_{i=1}^{10} E(V_i)$, and report the *average load* $((1/10)\Sigma_1^{10} E(V_i))$ of each simulation in (i.e., AVGLOAD), which ranges from 0.01 to 0.998.

The individual service requests were chosen from a uniform distribution in the following manner: Let $j = 1, \ldots, 100$ denote the test-case indices, and let $i = 1, \ldots, 10$ denote the individual station indices. We set $X_j = 0.01j$, $j = 1, \ldots, 100$ and then chose numbers A_j, B_j randomly generated from a uniform distribution over $(X_j - d_j, X_j + d_j)$, pairwise ordered so that $A_j < B_j$, $j = 1, \ldots, 100$. The individual service times at station i for the jth test case was distributed uniformly on (L_{ij}, U_{ij}), where L_{ij}, U_{ij} were themselves chosen randomly from a uniform (A_j, B_j) distribution, $i = 1, \ldots, 10$, $j = 1, \ldots, 100$. Note that $E(V_{ij}|A_j, B_j) = (A_j + B_j)/2$.

The grid size d_j was

$$d_j = \begin{cases} 0.009 & j = 1, \ldots, 4, \\ 0.04 & j = 5, \ldots, 8, \\ 0.07 & j = 9, \ldots, 12, \\ 0.1 & j = 12, \ldots, 100. \end{cases}$$

Although we wanted to keep a constant grid size across the entire scale, it was necessary to shorten the grid length at the very low (i.e., light traffic) loads to keep service times positive.

The cost parameters for each test case (i.e., h_{ij}, b_{ij}) were always chosen randomly from a uniform (0, 20) distribution.

Each test was simulated 10 times each under the two alternative operating disciplines, greedy (i.e., the index rule heuristic described above) and simple cyclic polling. Every simulation had run length of 1,000. For each run the total cost incurred under greedy was divided by the total cost under cyclic to get the *ratio*, which we use as our basic comparative measure. Averages and standard deviations were then calculated over the 10 runs. The *raw* data from each run (i.e., $100 \cdot 10 = 1000$ observations) is presented in Figure 1, as a plot of the Ratio $\cdot$ 100 against the average loads.

It is apparent from Figure 1 that only under very light loads (e.g., AVGLOAD from 0.01–0.06) is there no clear domination of greedy over cyclic. The greatest improvement seems to occur between average loads of 0.15–0.3 (i.e., for $1.5 \leqq \Sigma_{i=1}^{10} EV_i \leqq 3$), after which the ratio appears to increase towards 100% again.

The *summary* data (100 observations) is presented in a plot of the *average ratios* (of the 10 replications for each test case with identical parameters) against the average loads (i.e., XRATIO versus AVGLOAD) in Figure 2.

The identical plot is presented with 95% confidence intervals in Figure 3. As Figure 3 demonstrates, no upper confidence limit exceeds 100% after 0.06. The simulations suggest that the greedy heuristic (based on the index rule) outperforms cyclic polling under all but extremely light loads.

6. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

The indices derived above bear a strong resemblance to the important results of Harrison [7], which were later found to correspond to the Gittins index for Whittle's "open bandit process" (see Whittle [14, pp. 228–232]). The problem Harrison considered was setting the priorities for an asymmetric multiclass $M/G/1$ queue with infinite buffers to maximize total *discounted* reward over an infinite horizon. In Harrison's notation, the problem is equivalent to a terminal reward of $r_i + h_i/\beta$ for each type i served, where β denotes the discount factor.

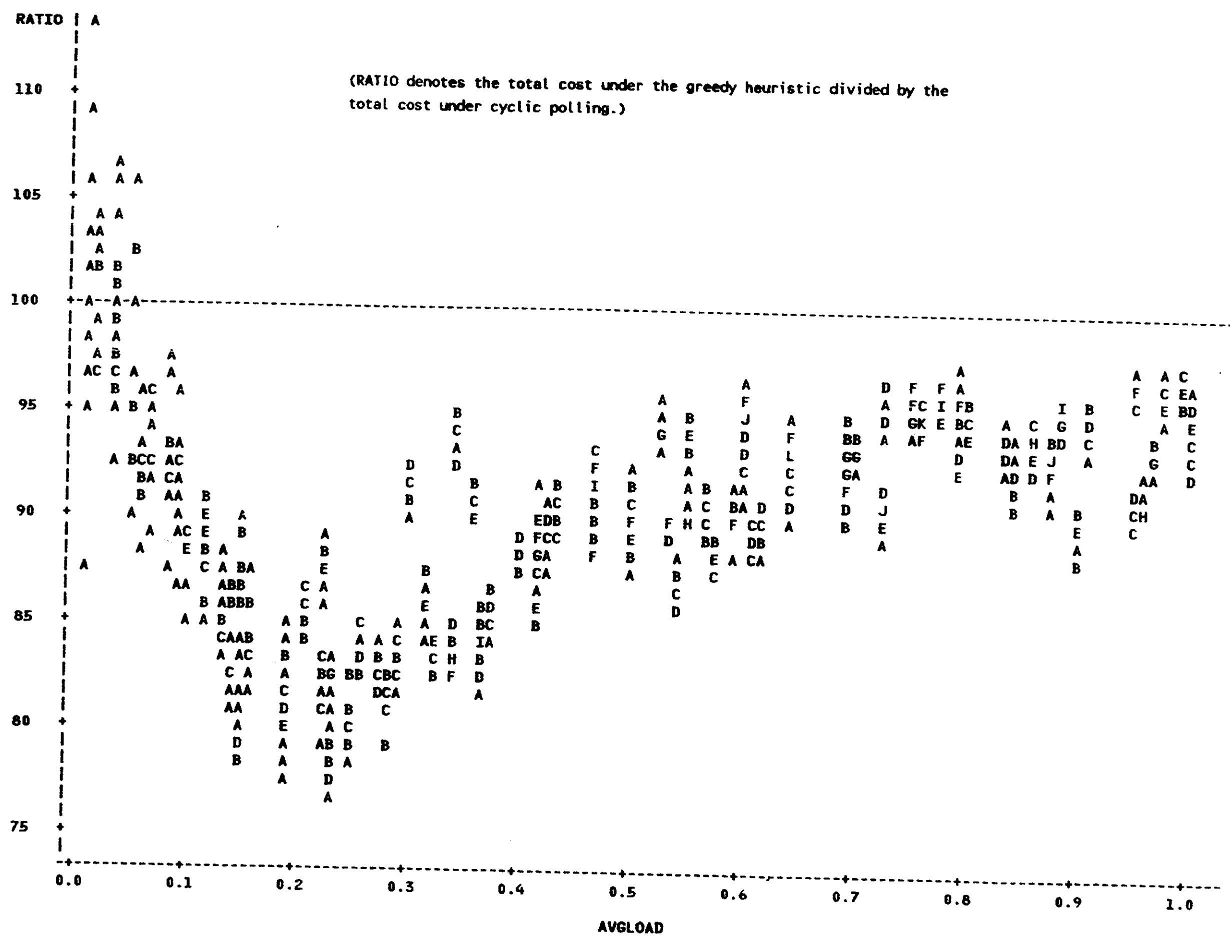

Figure 1. Simulation analysis: greedy versus cyclic. Plot of RATIO*AVGLOAD. Legend: A = 1 OBS, B = 2 OBS, etc.

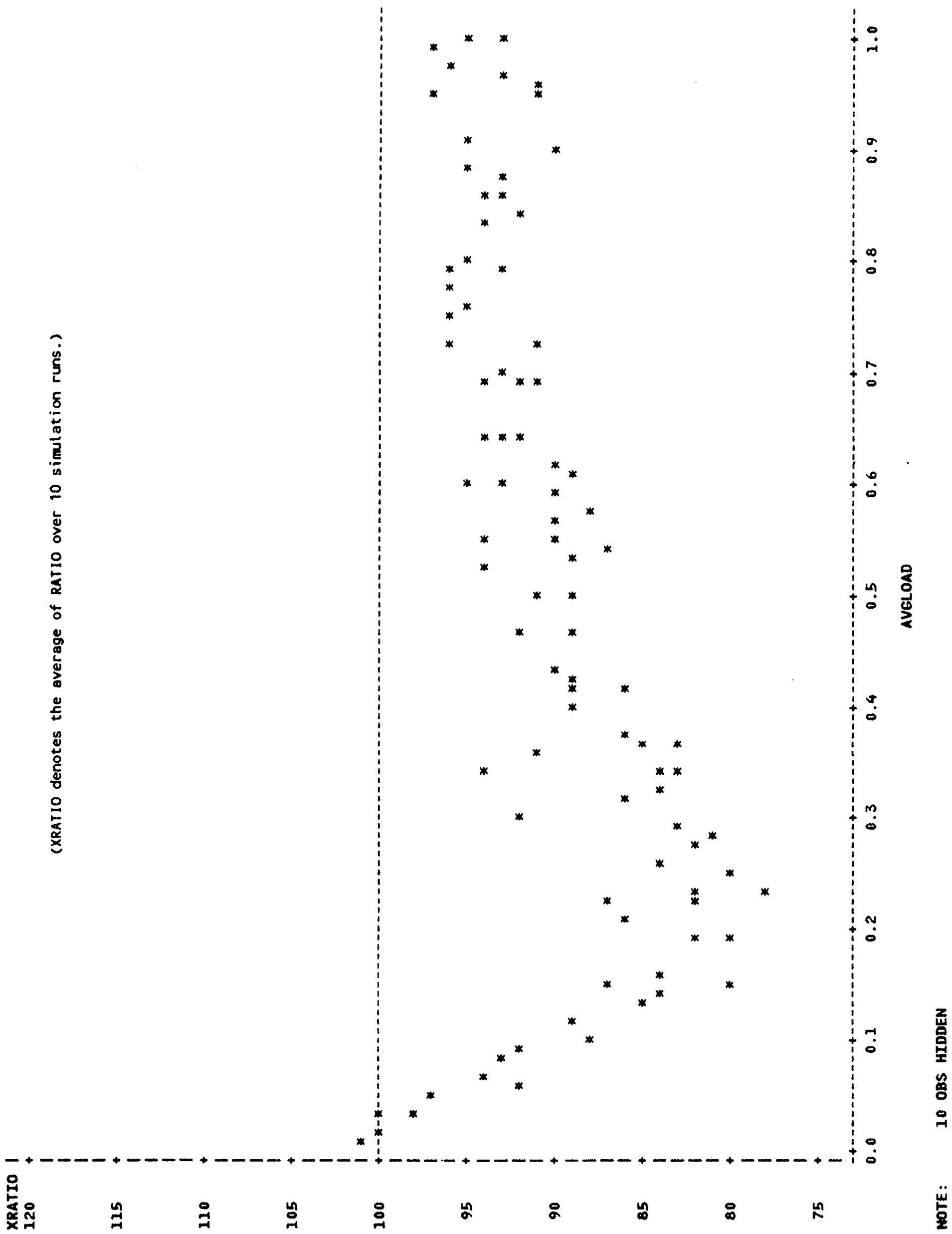

Figure 2. Means of runs of identical parameters. Plot of XRATIO*AVGLOAD. Symbol used is *.

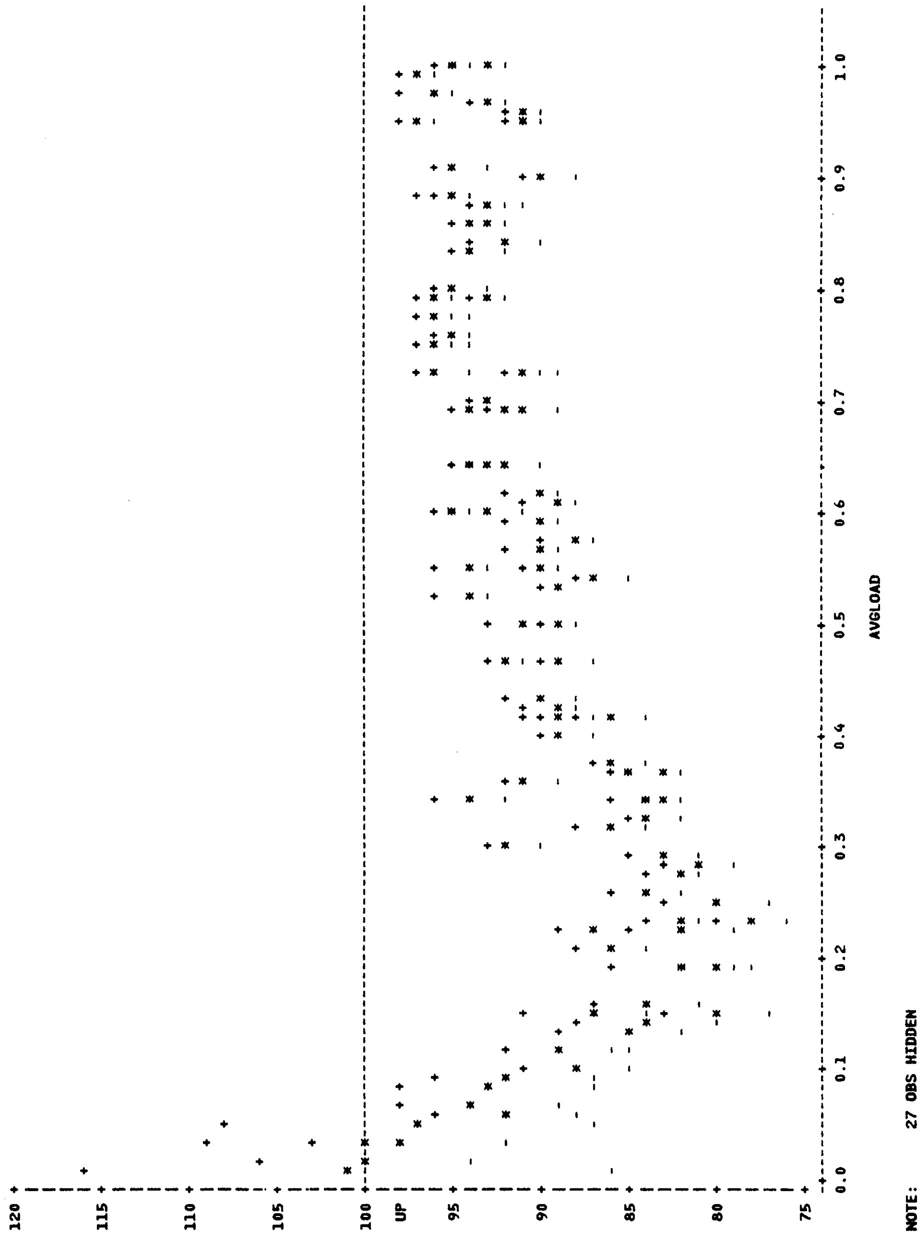

Figure 3. Confidence intervals for runs of identical parameters. Plot of UP*AVGLOAD, symbol used is *. Plot of DOWN*AVGLOAD, symbol used is -. Plot of XRATIO*AVGLOAD, symbol used is *.

value of the index

$$\frac{(r_i + h_i/\beta)\tilde{V}_i(\beta)}{1 - \tilde{V}_i(\beta)}$$

should be granted highest priority and gave the (complex) forms for the indices for the descending priorities.

Whittle [14] generalized Gittins [6] results on the multiarmed bandit to the case where new "arms" are generated via a homogeneous Poisson stream—the open bandit process—and proved that an index policy obtains discount optimality and rederived Harrison's results in that vein. Note that as $\beta \searrow 0$, Harrisons' index corresponds to the $c\mu$ rule $h_i/E(V_i)$ (here we call it $c\mu$, as arrivals are now still allowed, while previously our use of W.S.E.P.T. was more appropriate, as there no arrivals were allowed).

The key bandit theorems prescribe the policy of always serving the available customer with maximal index. This would seem to correspond to the (undiscounted though) one-step look-ahead policy described above; however, due to the unit buffers, the arrival stream (of serviced customers) can no longer be considered Poissonian. Whittle [15] recently conjectured on a bound for the distance from the optimal policy of the Gittins index policy for "restless bandits," where the arms may change in a Markovian manner when not being served. Although such a transformation is not known at this time, it would be extremely interesting to see if the system can be modeled as a restless bandit and use this result to clarify how close to optimality use of Theorems 1 and 2 is. This kind of approach may also yield additional insight into good operating policies for the fully asymmetric system.

Note Added in Proof: After submission of this paper, Hirayama, T. "Optimal Service Assignment In A Finite-Source Queue," *I.E.E.E. Trans. Aut. Contr.* **34**, 67–75 (1989), appeared where the optimal *preemptive* policy in the semi-symmetric case with zero switching times is obtained for the special case of *exponential* service times. The index obtained there reduces essentially to our eq. (12) for average cost optimality in that special case.

ACKNOWLEDGMENTS

The first author's research was funded by a grant from the Columbia University Graduate School of Business. Both authors would like to thank Robert Waitman for performing the simulations in Section 5. We would also like to acknowledge helpful comments and suggestions from the Associate Editor.

REFERENCES

[1] Baker, J.E., and Rubin, I., "Polling with a General-Service Orider Table," *IEEE Transactions on Communication*, **35**(3), 283–288 (1987).

[2] Browne, S., "Optimal Dynamic Operating Policies for Cyclic-Type Queues," unpublished dissertation, New York University, 1988.

[3] Browne, S., and Yechiali, U., "Dynamic Priority Rules for Cyclic Type Queues," *Advances in Applied Probability*, **21**(2) 432–450 (1989).

[4] Coffman, E.G., and Gilbert, E.N., "Polling and Greedy Servers on a Line," *Queueing Systems*, **2**, 115–145 (1987).
[5] Conway, R.W., Maxwell, W.L., and Miller, L.W., *Theory of Scheduling*, Addison-Wesley, Reading, MA, 1967.
[6] Gittins, J.C., "Bandit Processes and Dynamic Allocation Indices," *Journal of the Royal Statistical Society, Series B*, **41**, 148–164 (1979).
[7] Harrison, J.M., "Dynamic Scheduling of a Multi-Class Queue," *Operations Research*, **23**, 270–282 (1975).
[8] Katehakis, M.N., and Derman, C., "Optimal Repair Allocation in a Series System," *Mathematics of Operations Research*, **9**(4), 615–623 (1984).
[9] Nash, P. and Weber, R.R., "Dominant Strategies in Dynamic Allocation and Scheduling Problems," *Deterministic and Stochastic Scheduling*, M.H.A. Dempster, Ed., Reidel, Dordrecht, 1982.
[10] Ross, S.M., *Applied Probability Models with Optimization Applications*, Holden Day, San Francisco, 1970.
[11] Takagi, H., *Analysis of Polling Systems*, MIT Press, Cambridge, MA, 1986.
[12] Takagi, H., "Queuing Analysis of Polling Models," *A.C.M. Computing Surveys*, **20**, 5–28 (1988).
[13] Takine, T., Takehashi, and Hasegawa, T., "Exact Analysis of Asymmetric Polling Systems with Single Buffers," *IEEE Transactions on Communications*, **36**(10), 1119–1127 (1988).
[14] Whittle, P., *Optimization Over Time*, Wiley, New York, 1982, Vol. 1.
[15] Whittle, P., "Restless Bandits: Activity Allocation in a Changing World," *A Celebration of Applied Probability*, J. Gani, Ed., *Journal of Applied Probability*, Special Volume **25A**, 287–298.

Manuscript received November, 1988
Revised manuscript received October 5, 1990
Accepted October 24, 1990