



Proceedings
of The Third Yale Workshop on
Applications of Adaptive Systems Theory

Center for Systems Science
Becton Center
Yale University

June 15-17, 1983

A LEARNING APPROACH TO PRIORITY ASSIGNMENT IN A TWO CLASS
M/M/1 QUEUEING SYSTEM WITH UNKNOWN PARAMETERS

M. R. Meybodi and S. Lakshmivarahan

School of Electrical Engineering and Computer Science
University of Oklahoma
Norman, Oklahoma

ABSTRACT

A queueing system with two classes into which jobs arrive at constant Poisson but unknown rates and whose service times are exponentially distributed with unknown mean is considered. A learning approach to the problem of priority assignment in such a system is described. It is shown that by proper choice of the (step length) parameter of the learning algorithm, asymptotically, the class with the smaller average service time will be given higher priority with a probability as close to unity as desired.

INTRODUCTION

Priority assignment is an important topic in the study of queueing systems and has a number of interesting applications. For example, in packet switching computer networks acknowledgement messages are often given higher priority compared to regular data packets. In a centralised computer system in a multiprogramming environment, programs requiring shorter execution time are given higher priority compared to those requiring longer execution time. In a queueing system with priority assignment, jobs with different priority arrive and wait for service within each priority class. The priority assignment may either be preemptive or nonpreemptive. The now classical result on priority assignment states that if the highest priority is assigned to the class with the shortest average service time, then the overall mean waiting time for any job is less compared to the mean waiting time that would be obtained if the priority scheme were not used [7] [8] [12]. Thus, in order to develop a meaningful priority scheme, we need to know the probability characteristics of the jobs in various classes such as the arrival and service time distributions. An interesting question in this connection is that, if the probabilistic (arrival and service time) characteristics of jobs in various classes are not known in advance, how to go about developing a meaningful priority assignment among the various classes.

More formally, consider two classes of jobs each with Poisson arrival rate $\lambda_i (>0)$ and service time exponentially distributed with mean $\mu_i^{-1}, (>0), i=1,2$. There is only one server, no preemption

and the queue length in each class is allowed to be infinite. Within each class the service is on a first come first served basis. The following assumption is very crucial in our analysis.

(A1) The parameters λ_i and μ_i^{-1} , $i = 1, 2$ are not known. With loss of generality let $\mu_1^{-1} < \mu_2^{-1}$.

In this paper, we propose a learning approach to this problem of priority assignment when the probabilistic characteristics of the jobs in the two classes are not known in advance.

Varshavskii, Meleshina and Tsetlin [2] first considered this problem and proposed a solution using the so called fixed structure learning automaton algorithm where one automaton is used for each class. This latter class of algorithms were earlier developed by Tseltlin [1] in the context of modelling of collective behaviour. The fixed structure automaton is essentially an up-down counter. The current reading of the counter is called the state. The state transitions are indirectly controlled by the service requirements of jobs in each class. At any given instance the class corresponding to the automaton with highest state is chosen for service. This approach leads to a Markov chain with very complex transition structure which does not lend itself to formal analysis. In view of this difficulty Varshavskii et al [2] resorted to simulation to evaluate the effectiveness of this approach. They showed through extensive simulation that if the counter sizes (same as the number of states of each automaton) is chosen to be large to start with, their algorithm would asymptotically approach the priority assignment that would be used if the probabilistic characteristics were to be known.

In this paper using a class of learning algorithms very similar to those that are used in Mathematical Psychology [3] [4] [5] [9] we formally show that a randomised choice of classes for service would indeed asymptotically converge to the now classical priority assignment rule with a probability as close to unity as desired. In section 2 we describe the algorithm and state our main result. A proof of this theorem is presented in section 3. A number of interesting variations of this algorithm are presented in section 4.

A LEARNING ALGORITHM AND STATEMENT OF PROBLEM

The priority assignment queueing system considered in this paper is represented schematically in figure 1. Jobs arrive into two queues independently at constant Poisson rates $\lambda_i (>0)$ $i = 1, 2$. The service time requirement for jobs in class i and j (\neq) are independent. Further, within each class the service times are independent and are identically distributed with exponential distribution having mean μ_i^{-1} $i = 1, 2$. Let $\gamma^{(i)}$ be the random service time of a typical job from class $i = 1, 2$. Consider a stage when k_1 jobs from class i have been served. Then, $k = k_1 + k_2$ is the total number of jobs served by the system. Define a dynamic threshold

$$T(k) = \lambda_i [T^{(1)}(k_1) + T^{(2)}(k_2)]$$

where

$$T^{(i)}(k_i) = \frac{1}{k_i} \sum_{j=1}^{k_i} \gamma^{(i)}(j), T^{(i)}(0) = 0$$

and $\gamma^{(i)}(j)$ is the random service time for the j^{th} job in the i^{th} class.

A Dichotomy of Service Characterisation

It is assumed that class i has a service time distribution with exponential density function

$$f_i(t) = \mu_i^{-1} e^{-\mu_i t}$$

Given that k jobs have been given service, $k = 1, 2, 3, \dots$, define

$$d_i(k) = \text{Prob}[\gamma^{(i)}(k+1) < T(k)] = 1 - e^{-\mu_i T(k)}$$

and

$$c_i(k) = \text{Prob}[\gamma^{(i)}(k+1) > T(k)] = e^{-\mu_i T(k)}$$

Since $T(k)$ is a random variable, so is $d_i(k)$ $i=1, 2$. Clearly, $d_i(k)$ [$c_i(k)$] is the probability that the k^{th} job will have its service time less (more) than $T(k)$ if that job is from class i . Define $d(k) = (d_1(k), d_2(k))$. Notice $0 < d_i(k) < 1$ with probability one for $i=1, 2$ and all k . Further, since $\mu_1^{-1} < \mu_2^{-1}$ (assumption A1) it is easily seen that $d_1(k) > d_2(k)$ (1) with probability one for all k .

A Learning Algorithm

Let $p_i(k)$ be probability with which the DD in figure 1 decides to pick the $(k+1)^{\text{th}}$ job for service from class j . Define $p(k) = (p_1(k), p_2(k))$ where $p_1(k) + p_2(k) = 1$ and $k > 0$. Initially, $p_1(0) = p_2(0) = \frac{1}{2}$ with probability one.

Step 1: Let i be the class from which the $(k+1)^{\text{th}}$ job is chosen for service as a sample realisation from $p(k)$ for $k > 0$. At this instant, if the queue of this chosen class is non-empty, then select the job in the front of the queue for service. Otherwise, choose the job in the front of the other queue. Complete, the service and note $\gamma^{(i)}(k+1)$.

Step 2: The update timing algorithm: At the

instant when the class i is chosen for service (in step 1 as a sample realisation from $p(k)$) if the queues corresponding to both the class i as well as that of class $j (\neq i)$ are both non-empty then update $p(k)$ using step 3. Otherwise $p(k+1) = p(k)$ and go to step 4.

Step 3: Update Algorithms for $p(k)$: If the class i is chosen for service then

$$\begin{aligned} p_1(k+1) &= p_1(k) + \delta(1-p_1(k)) && \text{if } \gamma^{(i)}(k+1) \leq T(k) \\ p_j(k+1) &= p_j(k) - \delta p_j(k), j \neq i \\ \text{and} \\ p(k+1) &= p(k) && \text{if } \gamma^{(i)}(k+1) > T(k) \end{aligned} \quad (2)$$

where $0 < \delta < 1$ is called step length parameter.

Step 4: Update $T(k)$ using $\gamma^{(i)}(k+1)$.

Step 5: Go to step 1 until one of the components of $p(k)$ is unity. The following restrictive assumption is needed to simplify the analysis.

- A2. Except for the actual service needed by the jobs in step 1, all the other overhead operations such as the decision to pick a class from $p(k)$, checking whether the two queues are nonempty, the updating of $p(k)$ and $T(k)$ are all instantaneous.

In other words, there is no time delay involved in these overhead operations. Obviously this is a restrictive assumption. However, by employing fast micro processors, the overall overhead operation can be made very small, if not zero. We now state our main result.

Theorem 1: Under the assumptions A1 and A2, if $p(k)$ evolves according to the above learning algorithm, then for every $\epsilon > 0$, there exists a δ^* such that $0 < \delta^* < 1$ and for all $0 < \delta < \delta^*$

$$\lim_{k \rightarrow \infty} p_1(k) = 1 \geq 1 - \epsilon.$$

Stated in words the above theorem asserts that the above learning algorithm would evolve to the now classical assignment rule with a probability as close to unity as desired.

Obviously, there are other possible choices for both the update time and update algorithms. We shall indicate some of the other interesting variations in a later section. The above update algorithm has been extensively studied in the context of Mathematical Psychology [3][4] and in the context of learning automata [5][9]. Very recently similar learning algorithms have been applied to the two-person zero sum games [13], two person decentralised team [14] and decentralised routing in telephone networks [15].

PROOF OF THE MAIN RESULT

The proof of theorem 1 is presented in various steps. Let $p_i(k)$ be the probability that the queue of the class i is nonempty at the time when the $(k+1)^{\text{th}}$ job is chosen for service. Clearly $p_i(k)$ depends on λ_i , μ_i $i=1, 2$ and on $p(s)$, $0 \leq s \leq k$ and from A1 it follows that $0 < p_i(k) < 1$ with

probability one for all $k > 0$ and $i = 1, 2$. Let $\pi_1(k)$ be the probability with which $p(k)$ is updated.

Then

$$\pi_1(k) = p_1(k)p_2(k)[p_1(k)d_1(k) + p_2(k)d_2(k)] > 0$$

Define

$F_k = \sigma(p(0), p(0), d(1); p(1), d(2); \dots, p(k), p(k))$ be the smallest Borel-field [11] generated by the indicated random variables. That is F_k contains all the information regarding the choices of classes in step 1 states of the queue in step 2 and the results of the comparison of the service time with the dynamic threshold up to the instant when the $(k+1)^{th}$ job has been given service. Let $\Delta p_1(k) = E[p_1(k+1)|F_k] - p_1(k)$.

The following lemma is of immediate consequence.

Lemma 1: $\Delta p_1(k) \geq 0$ for all $k > 0$ with equality holding only when $p_1(k) = 0$ or 1.

Proof: By routine computation we obtain

$$\Delta p_1(k) = \pi_1(k)p_2(k)p_1(k)p_2(k)[d_1(k) - d_2(k)]$$

The lemma follows from the properties of $d_1(k)$ given in (1).

Corollary 1: $\lim_{k \rightarrow \infty} p_1$ exists and $\in \{0, 1\}$ with probability one.

Proof: Lemma 1 implies that $\{p_1(k)\}$ is a submartingale [11]. By the martingale theorem since $\{p_1(k)\}$ is non-negative and uniformly bounded $\lim_{k \rightarrow \infty} p_1(k) = p_1^*$ exists with probability one. Further, if $p_1(k) \notin \{0, 1\}$ then $p_1(k+1) \neq p_1(k)$ with non zero probability for all k . Hence $p_1^* \in \{0, 1\}$ which constitutes the absorbing set for the process $\{p_1(k)\}$.

Define

$$x_{p_1}$$

$$\Psi[p, x, \theta] = e^{\frac{x}{\theta}} \quad \text{where } x \text{ is real}$$

and

$$V[u] = \begin{cases} \frac{e^u - 1}{u} & \text{if } u \neq 0 \\ 1 & \text{if } u = 0 \end{cases}$$

It can be seen that $V[u]$ is non negative increasing and a convex function [3][5].

Lemma 2: There exists a positive, real number y such that

$$E[\Psi[p(k+1), y, \theta] | F_k] \leq \Psi[p(k), y, \theta].$$

Proof: By direct computation we obtain

$$E[\Psi[p(k+1), x, \theta] | F_k] = \Psi[p(k), x, \theta] - x p_1(k) p_2(k) \pi_1(k) \Psi[p(k), x, \theta] F[p(k), x]$$

where

$$F[p(k), x] = V[-x(1-p_1(k))] d_1(k) - V[x p_1(k)] d_2(k)$$

Lemma 2 follows if we can show that

$$F[p(k), x] > 0$$

that is, if

$$\frac{V[-x(1-p_1(k))]}{V[x p_1(k)]} \geq \frac{d_2(k)}{d_1(k)} \quad (3)$$

Defining $H[u] = \ln V[u]$ (where \ln stands for natural logarithm) and from the convexity properties of $H[u]$ it can be shown that [3][5][13]

$$\frac{V[-x(1-p_1(k))]}{V[x p_1(k)]} \geq \frac{1}{V[x]} \quad (4)$$

Combining these we see that (3) is true if

$$\frac{1}{V[x]} \geq n \text{ where } n = \sup_{k \in \mathbb{N}} \frac{d_1(k)}{d_2(k)} < 1. \quad (5)$$

From the properties of the function $V[u]$ it follows that there exists a real number $y > 0$ such that for all $x \in (0, y)$, inequality (5) will be true. By setting $x=y$, the lemma follows.

Now consider the function

$$h[z, x, \theta] = \frac{\frac{x}{\theta} z}{\frac{x}{\theta} - 1}$$

where $h[j, x, \theta] = j$ where $j = 0, 1$ and $0 < h[z, x, \theta] < 1$ for all $0 < z < 1$.

Lemma 3:

$$E[h[p_2(k+1), y, \theta] | F_k] \leq h[p_2(k), y, \theta] \quad (6)$$

where y is defined in lemma 2.

Proof: It can be seen that

$$h[p_2(k+1), y, \theta] = \frac{-\frac{y}{\theta} p_1(k) - \frac{y}{\theta}}{1 - e^{-\frac{y}{\theta}}} \quad (7)$$

Taking conditional expectations on both sides of (7), from the properties of conditional expectation and from lemma 2 the inequality (6) follows. Since $h[z, x, \theta]$ is continuous in z it follows that

$$\lim_{k \rightarrow \infty} h[p_2(k+1), y, \theta] = h[1 - p_1^*, y, \theta]$$

with probability one. Since $p_1^* \in \{0, 1\}$ with probability one we obtain $h[1 - p_1^*, y, \theta] = 1 - p_1^*$ (8) with probability one.

Proof with Theorem 1:

Taking expectations of both sides of (6), we obtain $E[h[p_2(0), y, \theta]] \geq E[h[p_2(1), y, \theta]] \geq E[h[p_2(2), y, \theta]] \geq \dots$

Thus

$$\begin{aligned} h[p_2(0), y, \theta] &\geq \lim_{k \rightarrow \infty} E[h[p_2(k), y, \theta]] \\ &= E[\lim_{k \rightarrow \infty} h[p_2(k), y, \theta]] \\ &= E[h[1 - p_1^*, y, \theta]] \\ &= E[1 - p_1^*] \\ &= \text{Prob}[p_1^* = 0] \end{aligned} \quad (9)$$

From the definition, it follows that $0 < z < 1, y > 0$

$$\lim_{\theta \rightarrow 0} h[z, y, \theta] = 0 \quad (10)$$

Now given $\epsilon > 0$, from (10) it follows that there exists a $\theta^*, 0 < \theta^* < 1$ such that for all $0 < \theta < \theta^*$

$$\epsilon > h[p_2(0), y, \theta] \quad (11)$$

Combining (9) and (11), $\epsilon > \text{Prob}[p_1^* = 0] = 1 - \text{Prob}[p_1^* = 1]$ and the theorem is proved.

DISCUSSION

1. Our method of proof of the theorem 1 is very similar to the one given in [10] in the context of the analysis of learning automata operating in a non-stationary environment as well as the one used in Chapter 5 in [5] for the analysis of the two person zero sum games with incomplete information.
2. When θ is small, the algorithm (3) is called the small step learning algorithm. This latter class of algorithms have been extensively studied by Norman [3] in the context of Mathematical Psychology. A major difference, however, is that in our analysis as in [10], we do not require $p(k)$ to be a Markov process.
3. Consider a variation of the update timing algorithm for $p(k)$.

Step 3: If the class chosen for service (as a sample realisation from $p(k)$ in step 1 is non-empty, then update $p(k)$ using step 4. Otherwise, $p(k+1) = p(k)$. The probability $d_2(k)$ with which $p(k)$ is updated according to this modified update rule is

$$a_2(k) = p_1(k)p_1(k)d_1(k) + p_2(k)p_2(k)d_2(k). \text{ Since } 0 < p_i(k) < 1 \text{ } i=1,2, \text{ it follows that } a_1(k) < a_2(k).$$

That is, according to this modified rule $p(k)$ is updated more frequently compared to original rule in step 3. Admittedly this step 3 could result in a faster convergence. However, it is not without further problems. We need an extra condition such as $p_1(k) > p_2(k)$ for the lemma 1 and 2 and hence theorem 1 to be true.

4. The algorithm (2) used in step 4 known as the linear reward-inaction algorithm[5], [9] is a special case of a more general absorbing barrier algorithms. One could easily generalize our results to these more general algorithms.

REFERENCES

- [1] M. L. Tsetlin. Automaton Theory and Modelling of Biological Systems. Academic Press 1973.
- [2] V.I. Varshavskii, M.V. Meleshina, M.L. Tsetlin. "Priority Organisation in Queueing Systems Using a Model fo the Collective Behavior of Automata." Problemy Perekhodchi Informatsii, Volume 4, 1968, pp. 73-76.
- [3] M.F. Norman. "On Linear Models with Two Absorbing Barriers." Journal of Mathematical Psychology, Vol. 5, 1968, pp. 225-241.
- [4] M.F. Norman. "Some Convergence Theorems for Stochastic Learning Models with Distance Diminishing Operators." Journal of Mathematical Psychology, Volume 5, 1968, pp. 61-101.
- [5] S. Lakshmivarahan. Learning Algorithms: Theory and Applications. Springer Verlag, 1981.
- [6] Yu.A. Flerov. "Some Classes of Multi-Input Automata." Journal of Cybernetics. Volume 2, 1972, pp. 112-122.
- [7] L. Kleinrock. Queueing Systems, Volume I and II. Wiley Interscience, 1975.
- [8] T.L. Satty. Elements of Queueing Theory, McGraw Hill.
- [9] K.S. Narendra and M.A.L. Thathachar. "Learning Automata-A Survey." IEEE Transactions on Systems Man and Cybernetics, Volume 4, 1974, pp. 323-334.
- [10] N. Baba and Y. Sawaragi. "On Learning Behaviour of Stochastic Automata under Non-Stationary Environment." IEEE Transactions of Systems, Man and Cybernetics.
- [11] J.L. Doob. Stochastic Processes. John Wiley 1951.
- [12] A. Cobham. "Priority Assignment in Waiting Line Problems." Operations Research. Volume 2, 1954, pp. 70-76.
- [13] S. Lakshmivarahan and K.S. Narendra. "Learning Algorithms for Two Person Zero Sum Stochastic Games with Incomplete Information." Mathematics of Operations Research. Volume 6, No. 4, 1981, pp. 379-386.

- [14] S. Lakshmivarahan. "Two Person Decentralized Team with Incomplete Information." Applied Mathematics and Computation. Volume 8, 1981, pp. 51-78.
- [15] P.R. Srikanta Kumar and K.S. Narendra. "Learning Algorithm Model for Routing in Telephone Networks." Becton Center SSIS Report 7903 Yale University, May 1979.
- [16] M.R. Meybodi and S. Lakshmivarahan. "A Learning Approach to Priority Assignment in a Two Class M/M/1 Queueing System with Unknown Paramters." School of EECS Technical Report, University of Oklahoma, January 1983.

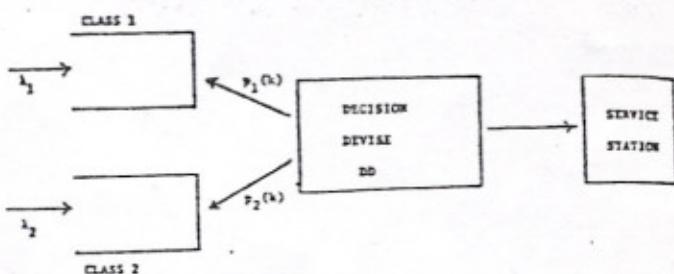


FIGURE 1.