SEQUENTIAL SCHEDULING OF PRIORITY QUEUES AND ARM-ACQUIRING BANDITS*

by

Chuanshu Ji
Purdue University
and
Columbia University

Technical Report 87–48

Department of Statistics
Purdue University

* Research partially supported by NSF grant DMS–8416736 and a David Ross fellowship.

SEQUENTIAL SCHEDULING OF PRIORITY QUEUES AND ARM-ACQUIRING BANDITS*

by

Chuanshu Ji

Department of Statistics
Purdue University
West Lafayette, IN 47907
USA

SUMMARY

In a queueing network with a single server and r service nodes, a non-preemptive non-idling policy chooses a node to service at each service completion epoch. Under the assumptions of independent Poisson arrival processes, fixed routing probabilities and linear holding cost rates, we apply Whittle's method for Arm-acquiring bandits to show that for minimizing discounted cost or long-run average cost the optimal policy is an index policy. We also give explicit expressions for those priority indices.

AMS Subject Classifications: Primary 90B22, 62L99; Secondary 90B15, 90B35.

Key words: sequential scheduling, queueing networks, Arm-acquiring bandits.

Running title: A Bandit Problem and Control of Queues.

* Research partially supported by NSF grant DMS–8416736 and a David Ross fellowship.

§1. Introduction

In this paper a queueing network consisting of a single server and r service nodes is considered. Each node allows an unbounded queue. At any time $t \geq 0$, service can only take place at one node (this is time-sharing service). The queueing discipline is non-preemptive and non-idling. The former requires that no interruption of service in a node is permitted. The latter means that the server can not be idle if at least one node has a non-empty queue. Here a queue includes any customer being serviced.

Several assumptions are made for the probability structure of the system:

(A1) The arrival process at node i from outside the network is a Poisson process with intensity λ_i, $i = 1, \ldots, r$. The r arrival processes at different nodes are independent.

(A2) The service times at node i are iid random variables, which need not have exponential distributions. The r service time sequences at different nodes are independent.

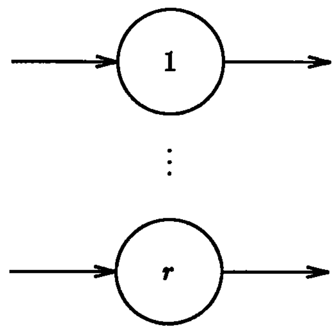
(A3) All service time sequences are independent of all arrival processes.

(A4) The service order at each node is "first-in-first-out". A customer who finishes his service at node i will either switch to the end of queue at node j with probability p_{ij}, or leave the network with probability $1 - \sum_j p_{ij}$.

(A5) The set of r nodes associated with a given partial order $\prec$ generates an oriented graph $\mathcal{G}$. $\mathcal{G}$ is a forest consisting of one-root trees oriented towards the root. Hence $\mathcal{G}$ contains no closed loops and may be decomposed into connectivity components, each of which is a tree; each tree has one root and is oriented towards this root. The root of a tree is the maximal element with respect to other vertices of the same tree.

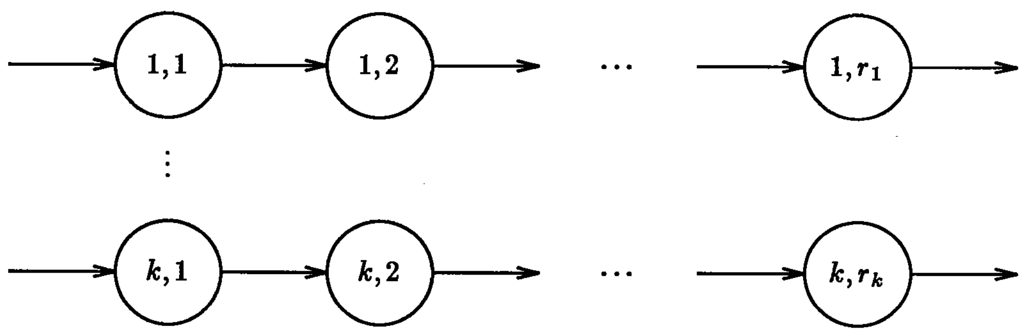
The order $\prec$ is defined as follows: Node j is said to be achievable from node i if there exist $n \in \mathbb{N}$ and nodes $i_1, \ldots, i_n$ such that $i_1 = i$, $\quad i_n = j$ and $p_{i_1 i_2} \cdots p_{i_{n-1} i_n} > 0$. We denote this by $p(i \to j) > 0$. Hence $i \prec j$ iff $p(i \to j) > 0$. Note that $p_{ij} > 0$ implies $i \prec j$ but the converse need not be true.

Example 1



Here $\lambda_i > 0$, $i = 1, \ldots, r$ and $p(i \to j) = 0$, if $i \neq j$.

Example 2



Here each node is coded by a pair (i, j): $j = 1, \ldots, r_i$, $i = 1, \ldots, k$. $r_1 + \ldots, + r_k = r$. Note that

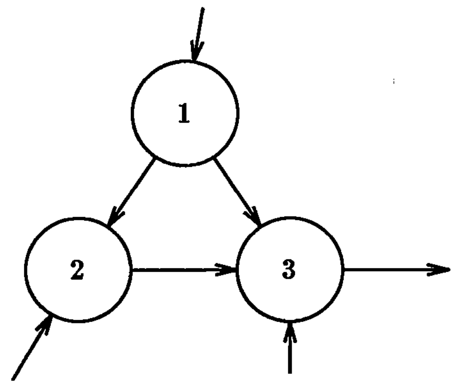
$\lambda_{i1} > 0, \quad \lambda_{ij} = 0, \quad j = 2, \ldots, r_i;$

$p_{(i,j),(i,j+1)} = 1, \quad j = 1, \ldots, r_i - 1;$

$p((i,j) \to (i',j')) = 0, \quad \text{if } i \neq i', \quad \text{or } i = i', \quad j = r_i, \quad \text{or } i = i', \quad j > j';$

$p((i,j) \to (i,j')) = 1 \quad \text{but } p_{(i,j),(i,j')} = 0, \quad \text{if } j + 2 \leq j' \leq r_i.$

Example 3



Here $r = 3$, $\lambda_i > 0$, $\quad i = 1, 2, 3$, $\quad$ and $p_{12} + p_{13} = 1$, $\quad p_{21} = 0$, $\quad p_{23} = 1$, $\quad p_{31} = p_{32} = 0$.

This queueing network is equivalent to a multi-class system with feedback probabilities, for one can view a customer at node i as a customer of type i, or simply an "i-customer", $i = 1, \ldots, r$. In what follows, we may refer to node i or i-customer, depending on which term is more convenient.

Now we introduce more notations. Let

$Q_i(t) =$ the queue length at node i and time t;

$c_i =$ the holding cost rate at node i;

$\tau_n =$ the nth service completion epoch;

$d_{n-1} =$ the node (code) which accepts service in the nth service stage $(\tau_{n-1}, \quad \tau_n)$;

Note that we usually choose node d_n at each epoch τ_n. However, if at τ_n all nodes have empty queues and the next new arrival at the network happens to be an j-customer, then $d_n = j$ automatically.

Every sequence $\{d_n, \quad n = 0, 1, \ldots\}$ specifies a policy π. For every $\alpha > 0$, define

$$V_{\alpha,\pi} = E^{\pi} \int_0^{\infty} e^{-\alpha t} \sum_{i=1}^{r} c_i Q_i(t) dt \quad ;$$

$$J_{\pi} = \liminf_{T \to \infty} \frac{1}{T} E^{\pi} \int_0^T \sum_{i=1}^{r} c_i Q_i(t) dt.$$

$V_{\alpha,\pi}$ is the expected total discounted cost with discount factor $e^{-\alpha}$ and policy π; J_{π} is the expected long-run average cost with policy π. In most cases of interest, the limit

$\lim_{T\to\infty} \frac{1}{T} E^\pi \int_0^T \sum_{i=1}^r c_i Q_i(t)dt$ actually exists.

Our goal is to find π_α and π^* such that

$$V_{\alpha,\pi_\alpha} = \inf_\pi V_{\alpha,\pi},$$

for every $\alpha > 0$; and

$$J_{\pi^*} = \inf_\pi J_\pi.$$

The problem of finding π_α was solved by Harrison [1] for the case of Example 1, using a direct policy improvement method. He also obtained π^* essentially in [2] for the same model. Following the same approach of Harrison with more elegant analysis, Tcha and Pliska [6] provided an algorithm for computing the optimal policy π_α for the general network model.

Klimov [3], [4] studied the general network model with the long-run average cost criterion. Assuming the system is in steady-state, he applied linear programming to characterize the optimal policy π^*.

Whittle [8] obtained the same results as in Harrison [1], [2], using the different method in [7], called *Arm-acquiring bandits (AAB)*. The bandit problem itself is very important, in which Whittle made a lot of contributions.

In this paper we investigate the general network model from the viewpoint of AAB. Motivated by Whittle's idea and methodology, we have succeeded in deriving explicit expressions for π_α and π^*. The two different fields — scheduling of priority queues and multi-armed bandits — have been tied together.

In section 2 the equivalence between our queueing problem and AAB is established by an adequate state-space transformation. We also state Whittle's results for AAB and give some heuristic explanations.

Section 3 contains the main results of this paper. To characterize the index policy π_α, we first derive a recursive formula for the priority indices, then apply the compound Poisson process theory to give the probability interpretation of those indices.

Based on the results of section 3, section 4 establishes the explicit expressions for π^*.

§2. Equivalence between sequential scheduling of priority queues (SSPQ) and Arm-acquiring bandits (AAB)

The problem given in section 1 may be called SSPQ. In this section we transform it to an equivalent problem of AAB.

Associated with each node i are the following traffic flows:

$A_i^O(t) = \#$ of arrivals at node i from outside the network in $[0,t]$;

$A_i^I(t) = \#$ of arrivals at node i from other nodes in $[0,t]$;

$D_i^O(t) = \#$ of departures from node i to outside the network in $[0,t]$;

$D_i^I(t) = \#$ of departures from node i to other nodes in $[0,t]$.

Then we have

$$Q_i(t) = Q_i(0) + A_i^O(t) + A_i^I(t) - D_i^O(t) - D_i^I(t). \tag{2.1}$$

Here we assume that all processes $\{Q_i(t)\}, \{A_i^O(t)\}, \{A_i^I(t)\}, \{D_i^O(t)\}, \{D_i^I(t)\}, \quad i = 1, \ldots, r$ have right-continuous realizations. Since

$$\int_0^t e^{-\alpha s} Q_i(s) ds = Q_i(0) \cdot \frac{1 - e^{-\alpha t}}{\alpha} + \int_0^t e^{-\alpha s} A_i^O(s) ds - \int_0^t e^{-\alpha s} [D_i^O(s) + D_i^I(s) - A_i^I(s)] ds,$$

we have

$$V_{\alpha,\pi} = C - \tilde{V}_{\alpha,\pi}, \tag{2.2}$$

where

$$\tilde{V}_{\alpha,\pi} = E^\pi \int_0^\infty e^{-\alpha t} \sum_{i=1}^r c_i [D_i^O(t) + D_i^I(t) - A_i^I(t)] dt,$$

and

$$C = E^\pi \{ \frac{1}{\alpha} \sum_{i=1}^r c_i Q_i(0) + \int_0^\infty e^{-\alpha t} \sum_{i=1}^r c_i A_i^O(t) dt \}.$$

It is observed that minimizing $V_{\alpha,\pi}$ is equivalent to maximizing $\tilde{V}_{\alpha,\pi}$ because C is actually a policy-independent quantity. Moreover, all the expectations $E^{\pi}(\cdot)$ are finite due to the following facts:

(i) $\max\{A_i^I(t), D_i^O(t), D_i^I(t)\} \le \sum_{i=1}^r A_i^O(t) + \sum_{i=1}^r Q_i(0), \quad \forall \quad i = 1,\dots,r, \quad t > 0;$

(ii) $0 \le E\int_0^\infty e^{-\alpha t} A_i^O(t)dt = \int_0^\infty e^{-\alpha t} E A_i^O(t)dt = \int_0^\infty e^{-\alpha t}\lambda_i t dt < \infty, \quad \forall \quad i = 1,\dots,r$

by the Monotone Convergence Theorem ;

(iii) We assume that $EQ_i(0) < \infty, \quad \forall \quad i = 1,\dots,r.$

Furthermore, assume that $D_i^O(0) = D_i^I(0) = A_i^I(0) = 0, \quad i = 1,\dots,r$. Since

$$\int_0^t e^{-\alpha s}[D_i^O(s) + D_i^I(s) - A_i^I(s)]ds = \frac{e^{-\alpha s}}{-\alpha}[D_i^O(s) + D_i^I(s) - A_i^I(s)] \quad \Big|_0^t + \frac{1}{\alpha}\int_0^t e^{-\alpha s}\cdot d[D_i^O(s) + D_i^I(s) - A_i^I(s)]$$

and

$$E\lim_{t\to\infty} e^{-\alpha t} A_i^O(t) = 0 \qquad \text{by Fatou's Lemma, we have}$$

$$\tilde{V}_{\alpha,\pi} = \frac{1}{\alpha} \quad \bar{V}_{\alpha,\pi}, \tag{2.3}$$

where

$$\begin{aligned}\bar{V}_{\alpha,\pi} &= E^{\pi}\int_0^\infty e^{-\alpha t}\sum_{i=1}^r c_i \cdot d[D_i^O(t) + D_i^I(t) - A_i^I(t)] \\ &= E^{\pi}\sum_{n=1}^\infty e^{-\alpha\tau_n}\sum_{i=1}^r c_i\cdot \Delta[D_i^O(t) + D_i^I(t) - A_i^I(t)]_{t=\tau_n}\end{aligned}$$

with the notation $\Delta[h(t)]_{t=t_0} \triangleq h(t_0^+) - h(t_0^-)$. Here we assume $\tau_0 = 0$ and observe that each random function $D_i^O(t) + D_i^I(t) - A_i^I(t)$ only has the jump points (up or down) at $\tau_n, \quad n \in \mathbb{N}$.

For every $\alpha > 0$ maximizing $\bar{V}_{\alpha,\pi}$ is a semi-Markov decision problem with state space

$$\mathcal{X} = \{q = (q_1,\dots,q_r) : \quad q_i = 0,1,2,\dots; \quad i = 1,\dots,r\}$$

and action space

$$\mathcal{A} = \{1,\dots,r\}.$$

Intuitively, q_i is the queue length at node i, and action i represents "servicing node i". Naturally we let $E_q^\pi(\cdot)$ denote the expectation given action i and state q.

A non-randomized Markov policy π is such a sequence $\{d_n, n = 0, 1, \ldots\}$ that every d_n depends only on the state at τ_n (or at the next new arrival epoch if all nodes have empty queues at τ_n). When d_n does not even depend on n, we call π a stationary policy. In this paper we omit definition of those more general policies (e.g. randomized, measurable, etc).

The dynamical programming equation for this problem is given by

Theorem 1. *For every $\alpha > 0$, there exists a stationary policy π_α such that $\bar{V}_{\alpha,\pi_\alpha} = \sup_\pi \bar{V}_{\alpha,\pi} \triangleq \bar{V}_\alpha$ and $\bar{V}_\alpha$ satisfies the equation*

$$\bar{V}_\alpha(q) = \max_{\substack{1 \le i \le r \\ q_i > 0}} L_i \bar{V}_\alpha(q), \tag{2.4}$$

where the one-stage operator L_i is defined by

$$\begin{aligned} L_i \bar{V}_\alpha(q) = {} & E_q^i[e^{-\alpha\tau} \sum_{j=1}^{r} c_j \cdot \Delta(D_j^O(t) + D_j^I(t) - A_j^I(t))_{t=\tau}] \\ & + \sum_{j=1}^{r} p_{ij} E_q^i[e^{-\alpha\tau} \bar{V}_\alpha(q^{(ij)} + w)] + (1 - \sum_{j=1}^{r} p_{ij}) E_q^i[e^{-\alpha\tau} \bar{V}_\alpha(q^{(i\cdot)} + w)], \end{aligned}$$

where τ is the generic notation for the duration of one service stage; $w = (w_1, \ldots, w_r)$ with w_i being the # of new arriving i-customers in the period τ;

$$q^{(ij)} = \begin{cases} (q_1, \ldots, q_j + 1, \ldots, q_i - 1, \ldots, q_r), & \text{if } i > j, \\ (q_1, \ldots, q_i - 1, \ldots, q_j + 1, \ldots, q_r), & \text{if } i < j, \end{cases}$$

represents that one customer moves from node i to node j; and

$$q^{(i\cdot)} = (q_1, \ldots, q_i - 1, \ldots, q_r)$$

represents that one customer leaves the network from node i. $q^{(ij)}$ and $q^{(i\cdot)}$ are well-defined for $q_i > 0$. Notice that $\bar{V}_\alpha(\cdot)$, called value function, depends on the initial state q in general.

Theorem 1 is a standard theorem of Blackwell type. For the proof, see Ross [5].

The problem of maximizing $\bar{V}_{\alpha,\pi}$ can be solved by using Whittle's AAB approach. To do that we need to introduce an additional action Δ, which stands for "retirement". At each epoch τ_n, we either choose some $i \in \mathcal{A}$ provided $Q_i(\tau_n) > 0$, or choose Δ with a constant welfare M. If $Q_i(\tau_n) = 0$ for all $i \in \mathcal{A}$, then Δ is the only choice. Once Δ is taken, service of the entire network will terminate from then on.

Let $\bar{V}_\alpha(q, M)$ be the analogue of $\bar{V}_\alpha(q)$ modified by adding action Δ with welfare M. Then the same conclusions as Theorem 1 hold for $\bar{V}_\alpha(q, M)$. We state them without proof as

Theorem 2. *For every $\alpha > 0$ and $M \in \mathbb{R}$, there exists a stationary policy $\pi_{\alpha,M}$ such that*

$$\bar{V}_{\alpha,\pi_{\alpha,M}} = \sup_{\pi} \bar{V}_{\alpha,\pi} \triangleq \bar{V}_\alpha$$

and $\bar{V}_\alpha$ satisfies the equation

$$\bar{V}_\alpha(q, M) = \max\{M, \quad \max_{\substack{1 \le i \le r \\ q_i > 0}} L_i \bar{V}_\alpha(q, M)\}. \tag{2.5}$$

The key point of AAB approach is to decompose (2.5) into r simultaneous equations, which are considerably easier to handle.

Let $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ be the state corresponding to a single i-customer. And let

$$\bar{V}_{i,\alpha}(M) = \bar{V}_\alpha(e_i, M),$$

$$E^i(\cdot) = E^i_{e_i}(\cdot),$$

$$A\Big(\frac{\partial \bar{V}}{\partial m}\Big) = E\Big[\prod_{i=1}^{r} \Big(\frac{\partial \bar{V}_{i,\alpha}(m)}{\partial m}\Big)^{A_i}\Big],$$

where A_i is a Poisson random variable with intensity λ_i, $i = 1, \ldots, r$. $\frac{\partial \bar{V}_{i,\alpha}(m)}{\partial m}$ is the usual notation for partial derivative, which will be justified later.

Theorem 3. (2.5) *is equivalent to the following r sumultaneous equations:*

$$\bar{V}_{i,\alpha}(M) = \max\{M, \quad L_i \bar{V}_{i,\alpha}(M)\}, \tag{2.6}$$

where $L_i\bar{V}_{i,\alpha}(M)$ *has the form*

$$L_i\bar{V}_{i,\alpha}(M) = (c_i - \sum_{j=1}^{r} c_j p_{ij})E^i e^{-\alpha\tau} + BE^i e^{-\alpha\tau}$$
$$- E^i\{\int_M^B \Big[\sum_{i=1}^{r} p_{ij} \cdot \frac{\partial \bar{V}_{j,\alpha}(m)}{\partial m} + (1 - \sum_{j=1}^{r} p_{ij})\Big]\Big[e^{-\alpha}A\Big(\frac{\partial \bar{V}}{\partial m}\Big)\Big]^{\tau} dm\},$$

where $i = 1, \ldots, r$ *and* $M \leq B < \infty$.

(2.6) here is just the analogue of [8], P227, (5), with the slight difference due to the greater generality of our network model. The verification can be done by repeating the argument in [7] with minor modification. For brevity we would rather make some heuristic remarks which emphasize more insight of Theorem 3.

Remarks:

(a) Starting with the initial state e_i, the one stage expected reward is given by

$$E^i[e^{-\alpha\tau}\sum_{j=1}^{r} c_j \cdot \Delta(D_j^O(t) + D_j^I(t) - A_j^I(t))_{t=\tau}] = (c_i - \sum_{j=1}^{r} c_j p_{ij})E^i e^{-\alpha\tau}. \quad (2.7)$$

In fact, given e_i we have

$$\Delta(D_j^O(t) + D_j^I(t))_{t=\tau} = \begin{cases} 1, & j = i, \\ 0, & j \neq i, \end{cases}$$

and $\Delta(A_j^I(t))_{t=\tau} = I_E$,

where E is the event that a customer finishing his service by τ will go to node j. Note that the transition probabilities p_{ij}, $i, j = 1, \ldots, r$ do not depend on τ, hence (2.7) holds.

(b) For every $\alpha > 0, M \in \mathbb{R}$, the optimal policy $\pi_{\alpha,M}$ is an index policy, which chooses certain node i with the largest priority index M_i provided $M_i > M$, where

$$M_i = \inf\{m \in \mathbb{R} : \quad \bar{V}_{i,\alpha}(m) = m\}, \quad i = 1, \ldots, r.$$

(c) The function $\bar{V}_{i,\alpha}(M), i = 1, \ldots, r$ are nondecreasing , convex and piecewise linear in M. Therefore, the derivatives $\frac{\partial \bar{V}_{i,\alpha}(m)}{\partial m}$ exist except at $m = M_j, j = 1, \ldots, r$. At those index points we may define them as the right-derivatives.

(d) Given a subset $\mathcal{B}$ of $\{1,\ldots,r\}$, π is said to be a *write-off policy* with *write-off set* $\mathcal{B}$ if π does not choose node i as the next service stage when $i \in \mathcal{B}$ at that decision epoch. If all nodes are written-off, then Δ will be the only available action. Obviously, the index policy $\pi_{\alpha,M}$ is a write-off policy with $\mathcal{B} = \{i : M_i \leq M\}$. Note that here $\mathcal{B}$ depends on M, denoted by $\mathcal{B}_M$. $\mathcal{B}_M \subset \mathcal{B}_{M'}$ when $M < M'$.

Start with small value M and let it increase. If we assume $M_1 \geq M_2 \geq \ldots \geq M_r$, then

$$\mathcal{B}_M = \begin{cases} \emptyset, & M < M_r, \\ \{j+1,\ldots,r\} & M_{j+1} \leq M < M_j, \quad j = 1,\ldots,r-1, \\ \{1,\ldots,r\}, & M \geq M_1. \end{cases}$$

Therefore $\mathcal{B}_M$ recruits new members when M passe the index points, and $\mathcal{B}_M$ keeps invariant when M lies between two adjacent index points.

(e) The free parameter M introduced in Theorem 2 and Theorem 3 seems to be a nuisance in the original queueing scheduling problem. However, it enables us to determime $M_1,\ldots,M_r$. Meanwhile, for sufficiently small M, $\pi_{\alpha,M}$ never chooses Δ unless at the decision epoch all nodes have empty queues. In that case $\pi_{\alpha,M}$ and π_α coincide.

§3. Construction of π_α

Following Whittle's notation, for every $\alpha > 0$ and $M \in \mathbb{R}$ we let $\phi_i(M) = \bar{V}_{i,\alpha}(M), i = 1,\ldots,r$. It is observed in section 2 that each $\phi_i(M)$ is a piecewise linear function and changes its slopes at each index point $M_j, j = 1,\ldots,r$. Therefore, if we find the slope of $\phi_i(M)$ on each piece (M_{j+1}, M_j), then those M_j's can be located as well. This idea is due to Whittle and can be carried out in our problem even the network structure is much more complicated.

Recall that M_i is the priority index of node i (or an i-customer). Assume that $M_1 \geq M_2 \geq \ldots \geq M_r$ since we can always number those nodes (or customers) in order of decreasing priority. For simplicity we also assume that $M_1 > M_2 > \ldots > M_r$, since $M_i = M_j$ means that node i and node j are equally preferable so that any tie breaker can be used.

For every $\alpha > 0$ and $i = 1, \ldots, r$, let

$$\psi_i(\alpha) = E^i e^{-\alpha \tau},$$

$$h_i = (c_i - \sum_{j=1}^{r} c_j p_{ij}) \psi_i(\alpha),$$

$$H_i(M) = B\psi_i(\alpha) - E^i \int_M^B \Big[\sum_{j=1}^{r} p_{ij} \frac{\partial \phi_j(m)}{\partial m} + (1 - \sum_{j=1}^{r} p_{ij})\Big] \Big[e^{-\alpha} \cdot A\Big(\frac{\partial \phi}{\partial m}\Big)\Big]^{\tau} dm,$$

where $M \leq B < \infty$. Then (2.6) becomes

$$\phi_i(M) = \max\{M, \quad h_i + H_i(M)\}. \tag{3.1}$$

Define

$$\psi_{i0} = 1, \qquad M > M_1;$$

$$\psi_{ij} = \frac{\partial \phi_i(M)}{\partial M}, \quad M_j > M > M_{j+1}, \quad j = 1, \ldots, r-1.$$

The next theorem gives a recursive formula for computing M_j's.

Theorem 4. *Consider a relabeling of nodes (or customers) at each decision epoch, so that node j has the j-th highest priority, $j = 1, \ldots, r$. Then having $M_1, \ldots, M_j$ determined, we have*

$$M_{j+1} = \max_{i \geq j+1} \quad \frac{\sum_{l=1}^{j} a_l M_l}{b_{ij}}, \tag{3.2}$$

where

$$a_l = [1 - \sum_{k=1}^{l-1} p_{ik}(1 - \psi_{k,l-1})] \cdot \psi_i(\alpha + \sum_{k=1}^{l-1} \lambda_k (1 - \psi_{k,l-1}))$$
$$- [1 - \sum_{k=1}^{l} p_{ik}(1 - \psi_{kl})] \cdot \psi_i(\alpha + \sum_{k=1}^{l} \lambda_k (1 - \psi_{kl})), \quad l = 1, \ldots, j;$$

and

$$b_{ij} = 1 - [1 - \sum_{k=1}^{j} p_{ik}(1 - \psi_{kj})] \cdot \psi_i(\alpha + \sum_{k=1}^{j} \lambda_k (1 - \psi_{kj})), \quad i \geq j+1, \quad j = 0, 1, \ldots, r-1.$$

Proof. Since

$$A\Big(\frac{\partial \phi}{\partial m}\Big) = exp\{-\sum_{j=1}^{r} \lambda_j(1 - \frac{\partial \phi_j(m)}{\partial m})\},$$

$$E^i[e^{-\alpha} A\Big(\frac{\partial \phi}{\partial m}\Big)]^r = \psi_i(\alpha + \sum_{k=1}^{r} \lambda_k(1 - \frac{\partial \phi_k(m)}{\partial m})).$$

Thus

$$\frac{\partial H_i(M)}{\partial M} = \Big[\sum_{k=1}^{r} p_{ik}\psi_{kj} + (1 - \sum_{k=1}^{r} p_{ik})\Big] \cdot \psi_i(\alpha + \sum_{k=1}^{r} \lambda_k(1 - \psi_{kj})),$$

$$for \quad M_j > M > M_{j+1}, \quad j = 0, 1, \ldots, r-1.$$

Set $B = M = M_1$, then $H_i(M_1) = M_1\psi_i(\alpha)$. Since

$$M_1 \geq h_i + H_i(M_1), \quad \forall \quad i = 1, \ldots, r,$$

with equality for i being assigned the label 1 in the new labeling, we obtain

$$M_1 = \max_{1 \leq i \leq r} \quad \frac{h_i}{1 - \psi_i(\alpha)}. \tag{3.3}$$

In general, having $M_1, \ldots, M_j$ determined,

$$\begin{aligned} &H_i(M_{j+1}) \\ &= M_1\psi_i(\alpha) - \int_{M_{j+1}}^{M_1} [\sum_{k=1}^{r} p_{ik} \frac{\partial \phi_k(m)}{\partial m} + (1 - \sum_{k=1}^{r} p_{ik})] \cdot \psi_i(\alpha + \sum_{k=1}^{r} \lambda_k(1 - \frac{\partial \phi_k(m)}{\partial m}))dm \\ &= M_1\psi_i(\alpha) + \sum_{l=1}^{j}(M_{l+1} - M_l)[\sum_{k=1}^{r} p_{ik}\psi_{kl} + (1 - \sum_{k=1}^{r} p_{ik})] \cdot \psi_i(\alpha + \sum_{k=1}^{r} \lambda_k(1 - \psi_{kl})) \\ &= M_{j+1}[\sum_{k=1}^{j} p_{ik}\psi_{kj} + (1 - \sum_{k=1}^{j} p_{ik})] \cdot \psi_i(\alpha + \sum_{k=1}^{j} \lambda_k(1 - \psi_{kj})) + \sum_{l=1}^{j} a_l M_l. \end{aligned}$$

The last step is due to the fact that $\psi_{kj} = 1$ for all $k \geq j+1$.

Since $M_{j+1} \geq h_i + H_i(M_{j+1})$ for all $i \geq j+1$, and the equality holds for i being assigned the label $j+1$ in the new labeling, (3.2) follows. □

(3.2) provides a recursive formula for computing the priority indices. However, for each $j = 0, 1, \ldots, r-1$, to calculate M_{j+1} we still need to know $\psi_{kl}, 1 \leq k \leq l \leq j$. Notice that ψ_{kl} is the slope of $\phi_k(M)$ on the piece (M_{l+1}, M_l). And it has very nice probabilistic interpretation.

For each $j = 0, 1, \ldots, r-1$, let

$$B_j = \{j+1, \ldots, r; \Delta\},$$

$$C_j = B_0 \backslash B_j.$$

In particular, $B_r = \{\Delta\}$, $C_0 = \emptyset$. Define

T_{kl} = the time needed to bring all relabeled i-customers ($i \in C_l$) to the set B_l

when the initial state is e_k, $1 \le k \le l \le r$.

Then we have

Proposition 1. $\psi_{kl} = E^k e^{-\alpha T_{kl}}$, $1 \le k \le l \le r$.

Proof. Given $M \in (M_{l+1}, M_l)$, $\pi_{\alpha,M}$ is a write-off policy with the write-off set B_l. Starting with the initial state e_k, $\pi_{\alpha,M}$ will service some node $i \in C_l$ in each stage until there is no i-customer ($i \in C_l$) in the network. Then $\pi_{\alpha,M}$ will retire and take the welfare M. Thus,

$$\phi_k(M) = V + ME^k e^{-\alpha T_{kl}},$$

where V is the expected reward before retirement, independent of M. Proposition 1 follows by differentiation. □

Notes.

(i) There is no presumption that $T_{kl} < \infty$. However our interest excludes the case that T_{kl} is a defective random variable, $1 \le k \le l \le r$. We impose the light-traffic condition, specified by

(*)
$$\rho = \sum_{i=1}^{r} \eta_i \mu_i < 1,$$

where $\mu_i = E^i \tau$ is the expected service time at node $i, i = 1, \ldots, r$; $\eta \triangleq (\eta_1, \ldots, \eta_r)'$ satisfies the traffic flow equations:

$$\eta_i = \sum_{j=1}^{r} p_{ji} \eta_j + \lambda_i, \quad i = 1, \ldots, r,$$

or in matrix form

$$(I_r - \mathbb{P}'(r))\eta = \lambda, \quad \lambda = (\lambda_1, \ldots, \lambda_r)',$$

where I_r is the $r \times r$ identity matrix, $\mathbb{P}(r)$ is the $r \times r$ matrix with entries $p_{ij}, i, j, = 1, \ldots, r$.

In fact, the assumption (A4) guarantees that $I_r - \mathbb{P}(r)$ is invertible hence η is uniquely determined. This will also be explained later in the proof of Lemma 2. Note that ρ is called the traffic intensity of the network and the condition (*) implies that T_{kl} has finite moments of any order, $1 \le k \le l \le r$.

(ii) T_{kl} depends on the target set B_l and the initial state e_k, but not on the order in which those nodes in the set C_l are serviced. In what follows, we apply compound Poisson process theory to derive the expressions of $E^k e^{-\alpha T_{kl}}, 1 \le k \le l \le r$.

Lemma 1. *Let Z be a non-negative continuous random variable satisfying $P(Z > 0) = 1$, $P(Z < 1) > 0$ and $EZ < \infty$. Then for every $\beta \in (0,1)$ the equation $e^{-u} = \beta E e^{-uZ}$ has a solution $u > 0$.*

Proof. Let $g(u) = \beta E e^{-uZ} - e^{-u}$, then

(i) $g(0) = \beta - 1 < 0$;

(ii) $\lim_{u \to \infty} g(u) = 0$, since $P(Z = 0) = 0$;

(iii) $g(u)$ is a continuous function for $u > 0$.

Since $EZ < \infty$ and Z is a continuous random variable, by the Dominated Convergence Theorem we obtain that

$$g'(u) = -\beta E(Ze^{-uZ}) + e^{-u} = e^{-u}[-\beta E(Ze^{-u(Z-1)} I_{(Z>1)}) - \beta E(Ze^{-u(Z-1)} I_{(Z<1)}) + 1].$$

Note that $\lim_{u \to \infty} E[Ze^{-u(Z-1)} I_{(Z>1)}] = 0$, and by Fatou's lemma,

$$\lim_{u \to \infty} E[Ze^{-u(Z-1)} I_{(Z<1)}] \ge E[\lim_{u \to \infty} Ze^{-u(Z-1)} I_{(Z<1)}] = \infty.$$

Therefore, $g'(u) < 0$ for sufficiently large u. Hence (i), (ii), (iii) imply that there exists a $u > 0$ such that $e^{-u} = \beta E e^{-uZ}$. $\square$

In queueing literature the term "workload" is usually referred to service time(s) associated with a customer. Even in this complex network model we can still imagine that each arriving customer brings certain workload, which is the sum of service times corresponding to those nodes along his route in the network. Let

X_k be the generic notation for the service time at node k, $k = 1, \ldots, r$;

Y_n be the workload brought by the n-th arriving customer at the network, $n \in \mathbb{N}$. (Here we assume that no more than one customers arrive at the network at same time.)

For every $j = 1, \ldots, r$, let

I_j = the $j \times j$ identity matrix;

$\mathbb{P}(j)$ = the $j \times j$ matrix with entries p_{kl}, $\quad k, l = 1, \ldots, j$;

$v(j) = (v_1, \ldots, v_j)'$ with $v_l = \frac{\lambda_l}{\lambda_1 + \ldots + \lambda_j}$, $\quad l = 1, \ldots, j$;

$X(j) = (X_1, \ldots, X_j)'$;

$U_l^{(j)}$ = the workload brought by a customer arriving at node l towards the target set B_j,

i.e. $U_l^{(j)}$ only includes those service times at nodes in C_j.

Lemma 2. *For fixed $j = 1, \ldots, r$, suppose the workload sequence $\{Y_n\}$ is defined with respect to the target set B_j, then $Y_1, Y_2, \ldots$ are iid random variables, and there exists a random variable Y such that*

(i) *Y and Y_1 have the same distribution;*

and

(ii) $Y = v'(j) \cdot (I_j - \mathbb{P}(j))^{-1} \cdot X(j)$.

Proof. Recall (A1), (A2), (A3) and notice that the transition probability matrix $\mathbb{P}(j)$ does not depend on any arrival process or service time sequence. So $Y_1, Y_2, \ldots$ are iid.

For an arbitrary arrival customer with workload Y, we have

$$Y = \sum_{l=1}^{j} v_l \cdot U_l^{(j)}. \tag{3.4}$$

Suppose he enters the network at node l. After time X_l he may reach the target set B_j with probability $1 - \sum_{i=1}^{j} p_{li}$, then no more workload is left with him. Or with probability p_{li} he goes to node i $(i \in C_j)$, then his updated workload is $U_i^{(j)}$. Therefore,

$$(3.5) \qquad U_l^{(j)} = \sum_{i=1}^{j} p_{li}(X_l + U_i^{(j)}) + (1 - \sum_{i=1}^{j} p_{li})X_l = X_l + \sum_{i=1}^{j} p_{li}U_i^{(j)}, \quad 1 \le l \le j.$$

In matrix form (3.5) is written as

$$(I_j - \mathbb{P}(j))(U_1^{(j)}, \dots, U_j^{(j)})' = X(j).$$

By (A4) every customer will reach the target set B_j after entering the network and passing through a finite number of nodes in C_j. This implies that $I_j - \mathbb{P}(j)$ is invertible (cf. Klimov [3], Lemma 3). Therefore,

$$(3.6) \qquad (U_1^{(j)}, \dots, U_j^{(j)})' = (I_j - \mathbb{P}(j))^{-1}X_{(}j).$$

(ii) follows by combining (3.4) and (3.6). □

Proposition 2. *Under the light-traffic condition* $(*)$, *for every* $\alpha > 0$ *we have the expression*

$$(3.7) \qquad E^k e^{-\alpha T_{kl}} = Ee^{-uX_k}, \quad 1 \le k \le l \le r,$$

where $u > 0$ *satisfies the equation* $e^{-u} = e^{-\alpha}Ee^{-uZ}$ *in Lemma 1 with*

$$(3.8) \qquad Ee^{-uZ} = exp\{-(\lambda_1 + \dots + \lambda_j)(1 - Ee^{-uY})\},$$

where Y *is defined by Lemma 2.*

Proof. Let

N_t = the total # of customers arriving at all nodes of C_j in $[0, t]$;

$S_t^{(j)}$ = the total residual workloads at time t with respect to the target set B_j. i.e. $S_j^{(j)}$ is the sum of workloads associated with all customers at those nodes of C_j and at time t.

Given the initial state e_k, we have

$$S_0^{(j)} = X_k; \tag{3.9}$$

$$S_t^{(j)} = X_k + \sum_{n=1}^{N_t} Y_n - t, \quad 0 \le t \le T_{kj}; \tag{3.10}$$

and

$$T_{kj} = \inf\{t \ge 0 : \quad S_t^{(j)} = 0\}. \tag{3.11}$$

Let $Z(t) = \sum_{n=1}^{N_t} Y_n$, then $\{Z(t), t \ge 0\}$ is a compound Poisson process. Consider the Laplace transform

$$h(u) = Ee^{-uZ(1)},$$

where u satisfies $e^{-u} = e^{-\alpha} h(u)$ (cf. Lemma 1). Since $[e^u h(u)]^t$ is the Laplace transform of $Z(t) - t$, we claim that

$$\{U(t) \triangleq \frac{e^{-u(Z(t)-t)}}{[e^u h(u)]^t}, \quad t \ge 0\}$$

is a martingale. By the light-traffic condition and the Optional Sampling Theorem, $\{U(0), \quad U(T_{kj})\}$ is a two-point martingale. So

$$1 = U(0) = E^k(U(T_{kj}) | S_0^{(j)}) = e^{uX_k} \cdot E^k(e^{-\alpha T_{kj}} | S_0^{(j)}),$$

or

$$E^k(e^{-\alpha T_{kj}} | S_0^{(j)}) = e^{-uX_k}.$$

Hence (3.7) follows.

To show (3.8), let $Z = Z(1)$. Then $Z = \sum_{n=1}^{N_1} Y_n$, where N_1 has Poisson distribution with intensity $\lambda_1 + \ldots + \lambda_j$. Because $\{N_t\}$ and $\{Y_n\}$ are independent, we have

$$Ee^{-uZ} = \sum_{k=0}^{\infty} P(N_1 = k) \cdot E\Big[exp\Big(-u \sum_{n=1}^{k} Y_n\Big)\Big] = exp\{-(\lambda_1 + \ldots + \lambda_j)(1 - Ee^{-uY})\},$$

where Ee^{-uY} can be computed by using Lemma 2 (ii). Hence (3.8) follows. □

So far we have completed the algorithm for computing indices $M_1, \ldots, M_r$.

§4. Construction of π^*

It usually happens that the optimal policy with respect to long-run average cost is the limit of the optimal policy for discounted cost as the discount factor tends to one. This is indeed the case between π^* and π_α.

Theorem 5. *Under the light-traffic condition* (*), π_α *will tend to* π^* *as* α *approaches zero.*

Proof. In this queueing network a busy period is counted from the first arrival epoch (after the server was idle) to the first time that all nodes have empty queues. Assuming light-traffic we have an alternating busy-idle sequence. Since only non-idling policies are considered, and all arrival processes and the transition matrix $\mathbb{P}(r)$ are policy-independent, it turns out that the duration of a busy period is policy-independent as well. And the successive busy periods form an iid sequence. The light-traffic condition also implies that a busy period has finite moments of any order. Then Theorem 5 follows from [5], section 7.4. □

For each $\alpha > 0$, π_α is characterized by the priority indices $M_1, \ldots, M_r$ in Theorem 4. To characterize π^*, we need to evaluate the asymptotic behavior of M_i's as α is close to zero.

Theorem 6. *Let*

$$\lim_{\alpha \to 0} \alpha M_j = \tilde{M}_j, \quad j = 1, \ldots, r.$$

Then π^* *is characterized by* $\tilde{M}_1, \ldots, \tilde{M}_r$,

$$\tilde{M}_{j+1} = \max_{i \geq j+1} \frac{\sum_{l=1}^{j} \tilde{a}_l \tilde{M}_l}{\tilde{b}_{ij}}, \tag{4.1}$$

where

$$\tilde{a}_l = \sum_{k=1}^{l} p_{ik} E^k T_{kl} + \mu_i \sum_{k=1}^{l} \lambda_k E^k T_{kl} - \sum_{k=1}^{l-1} p_{ik} E^k T_{k,l-1} - \mu_i \sum_{k=1}^{l-1} \lambda_k E^k T_{k,l-1};$$

and

$$\tilde{b}_{ij} = \mu_i + \sum_{k=1}^{j} E^k T_{kj}(p_{ik} + \lambda_k \mu_i), \quad i \geq j+1, \quad j = 0, 1, \dots, r-1.$$

Proof. As $\alpha \to 0$,

$$\psi_i(\alpha) = 1 - \alpha\mu_i + o(\alpha)$$

and

$$\psi_{kl} = 1 - \alpha E^k T_{kl} + o(\alpha).$$

Then $\tilde{a}_l$, $\tilde{b}_{ij}$ are derived by Taylor expanding a_l and b_{ij} in (3.2). □

To implement π^*, we still need to compute all $E^k T_{kl}$'s.

Proposition 3. *For every* $j = 1, \dots, r$, *let*

$\gamma(j) = (\gamma_1^{(j)}, \dots, \gamma_j^{(j)})$ *with* $\quad \gamma_l^{(j)} = EU_l^{(j)}, \quad 1 \leq l \leq j.$

Then

$$E^k T_{kj} = \frac{\mu_k}{1 - \sum_{l=1}^{j} \lambda_l \gamma_l^{(j)}}, \quad 1 \leq k \leq j. \tag{4.2}$$

Proof. First of all, since $(I_r - \mathbb{P}(r))^{-1}$ exists and

$$(I_r - \mathbb{P}(r))\gamma(r) = \mu, \qquad \mu = (\mu_1, \dots, \mu_r)',$$

we have

(i) $\gamma(r) = (I_r - \mathbb{P}(r))^{-1}\mu$

(ii) $\lambda = (I_r - \mathbb{P}'(r))\eta$ (the traffic flow equation).

(i), (ii) and (∗) imply that

$$\lambda'\gamma(r) = \eta'\mu < 1.$$

So $\quad 1 - \sum_{l=1}^{j} \lambda_l \gamma_l^{(j)} > 0 \quad$ for all $j = 1, \dots, r$.

Furthermore, applying Wald's identity to (3.10) we obtain that

$$0 = E^k S^{(j)}_{T_{kj}} = \mu_k + EN_{T_{kj}} \cdot EY - E^k T_{kj} = \mu_k + (\lambda_1 + \ldots + \lambda_j) E^k T_{kj} \cdot EY - E^k T_{kj}.$$

By (3.4), $EY = v'(j) \cdot \gamma(j)$. Hence (4.2) holds. □

Acknowledgement

The author wishes to express his gratitude to Professors Tze Leung Lai, Steven Lalley and Ioannis Karatzas for helpful discussions and valuable support.

References

[1] Harrison, J. M. (1975). Dynamic scheduling of a multiclass queue: discounted optimality. Opns. Res. 23, 270 – 282.

[2] Harrison, J. M. (1976). Dynamic scheduling of a twoclass queue: small interest rates. SIAM J. Appl. Math. 31, 51 – 61.

[3] Klimov, G. P. (1974). Time-sharing service system. I. Theory Prob. Appl. 19, 532 – 551.

[4] Klimov, G. P. (1978). Time-sharing service system. II. Theory Prob. Appl. 23, 314 – 321.

[5] Ross, S. (1970). *Applied probability models with optimization applications.* Holden-Day, San Francisco.

[6] Tcha, D. and Pliska, S. R. (1977). Optimal control of single-server queueing networks and multi-class M/G/1 queues with feedback. Opns. Res. 25, 248 – 258.

[7] Whittle, P. (1981). Arm-acquiring bandits. Ann. Prob. 9, 284 – 292.

[8] Whittle, P. (1982). *Optimization over time. Vol.I* Wiley, New York.

From ubi Wed Nov 18 13:16:13 1987
Received: by l.cc.purdue.edu (5.54/1.14)
        id AA16014; Wed, 18 Nov 87 13:16:11 EST
Date: Wed, 18 Nov 87 13:16:11 EST
From: ubi (ji)
Message-Id: <8711181816.AA16014@l.cc.purdue.edu>
To: tek
Status: RO

Teena:

I am sending you two files ``paper2.pici'' and ``Makefile''.

``Makefile'' contains the information about how to compile and print

``paper2.pici'', which is just my technical report 87-48. For simplicity,

to compile it just type ``make p'';

to print it type ``make pp''.

Thank you.

Ji 11/18/87

See Mark to run this.
It has pictures in it.