Proceedings of the 31st Conference
on Decision and Control
Tucson, Arizona • December 1992

Optimal Control of Arrivals to Queues with Delayed Queue Length Information

Joy Kuri and Anurag Kumar
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore-560012, INDIA
e-mail: jaykay@ece.iisc.ernet.in anurag@ece.iisc.ernet.in

Abstract

We consider discrete time versions of two classical problems in the optimal control of admission to a queueing system: (i) optimal routing of arrivals to two parallel queues and (ii) optimal acceptance/rejection of arrivals to a single queue. We extend the formulation of these problems to permit a k step delay in the observation of the queue lengths by the controller. For geometric inter-arrival times and geometric service times the problems are formulated as Controlled Markov Chains with expected total discounted cost as the minimization objective.

For problem (i) we show that when $k = 1$, the optimal policy is to allocate an arrival to the queue with the smaller *expected* queue length (JSEQ: Join the Shortest Expected Queue). For $k \geq 2$, however, JSEQ is not optimal.

For problem (ii) we show that when $k = 1$, the optimal policy is a *threshold* policy. There are, however, *two* thresholds $m_0 \geq m_1 > 0$, such that m_0 is used when the previous action was to reject, and m_1 is used when the previous action was to accept.

1 Introduction

We consider discrete time and delayed queue-length information versions of two classical problems ([1],[6],[8],[9]), for which explicit structural results have been obtained for the zero delay case in the above mentioned references.

The first problem is that of optimally allocating arriving customers to one of two parallel queues so as to minimise the expected total discounted number in the system. For exponential service times the optimality of the Join the Shortest Queue (JSQ) policy is the well known result for this problem ([9], [8], [1]). We consider geometric inter-arrival times and geometric service times. Further, we assume that the controller is located so that it can observe only queue lengths after a delay of k time steps. For $k = 1$, we show that the optimal policy is for the controller to calculate the expected queue lengths conditioned on the most recently known queue lengths and the controls applied since then, and then allocate an arrival to the queue with the smaller expected length, i.e., the policy is now JSEQ, Join the Shortest Expected Queue. We have also shown that for $k \geq 2$, JSEQ is no longer optimal.

The second problem is that of optimally accepting or rejecting customers arriving to a single queue so as to minimise the expected total discounted cost, where there is a fixed cost per unit time for each queued customer and there is a reward for each accepted customer. For general i.i.d. arrivals and exponential service times it has been shown that the optimal policy is of threshold type ([6]). We consider geometric inter-arrival times and geometric service times. Further, we assume that the controller is so located that it can observe only the queue length after a delay of k units. For $k = 1$, we show that the optimal policy is again of threshold type. There are *two* thresholds $m_0 \geq m_1 > 0$; if the action one step back was to reject (resp. accept) then the optimal policy accepts if the queue length one step back was less than m_0 (resp. m_1).

2 Optimal Customer Allocation to Two Parallel Queues

We assume that time t is discrete. Let $(q_1(t), q_2(t))$ denote the discrete time queue length process, where by "queue length" we mean the total number in the queue, including the service position. At time t, $t \in \{0, 1, 2, \ldots\}$ the controller must decide on a control action $u(t) \in \{1, 2\}$, and is allowed to observe only the queue lengths till time $t - k$ and, of course, knows all control actions till time $t - 1$. In particular, we assume that at time 0 the process has already been evolving since time $-k$ and the controller is given $(q_1(-k), q_2(-k))$ and $((u(-k), u(-k+1), \ldots, u(-1))$. The problem is to choose $\{u(0), u(1), \ldots\}$ so as to optimise a cost function.

The arrivals and departures occur as follows. An arrival occurs to the system with probability λ at $t = n+$, $n \geq -k$, and a departure occurs from a non-empty queue with probability μ at $t = n-$, $n \geq -k+1$. The control action at $t = n$, $n \geq -k$, decides to which queue an arrival at $n+$ must be routed. If no customer arrives at $n+$ then the decision has no effect.

The scheduler, at time t, has the information $\{\{q_i(t-l)\}_{l=k}^{t+k}, i = 1, 2\}$ and $\{u(t-l)\}_{l=1}^{t+k}\}$. We need a policy π for choosing $\{u(0), u(1), u(2) \ldots\}$ so as to minimise the following cost function: $E^{\pi}_{s(0)}[\sum_{n=0}^{\infty} \beta^n (\lambda + q_1(n) + q_2(n))]$, where $s(0) = \{\{q_i(-k)\}, i = 1, 2, \{u(-k), \ldots, u(-1)\}\}$ and $\beta \in (0,1)$ is a discount factor. It is clear that $(\lambda + q_1(n) + q_2(n))$ is the expected holding cost of customers in the interval $n \in \{0,1,2, \ldots\}$ if the holding cost per customer per time step is 1.

The problem posed above can be formulated directly as a CO-CMC (Completely Observed-Controlled Markov Chain).

We show the formulation here for $k = 1$. We list the elements of the CO-CMC for $k = 1$ as follows.

a) State at time n : $s(n) = (q_1(n-1), q_2(n-1), u(n-1))$ $\forall n \in \mathcal{N}$. So the state space is $\mathcal{N} \times \mathcal{N} \times \{1, 2\}$.

b) Action at time n : $u(n) \in \{1, 2\}$ $\forall n \in \mathcal{N}$.

c) Transition probabilities : Let $i = (i_1, i_2, v)$, $j = (j_1, j_2, w)$; $i_1, i_2, j_1, j_2 \in \mathcal{N}$, v, $w \in \{1, 2\}$. Then Prob.$(s(n+1) = j | s(n) = i, u(n) = d) = I\{w = d\} \times$ Prob.$(q_1(n) = j_1, q_2(n) = j_2 | q_1(n-1) = i_1, q_2(n-1) = i_2, u(n-1) = v)$.

We denote by P_v the $\mathcal{N}^2 \times \mathcal{N}^2$ matrix with elements Prob.$(q_1(n) = j_1, q_2(n) = j_2 | q_1(n-1) = i_1, q_2(n-1) = i_2, u(n-1) = v)$.

Consider a function $f : \mathcal{N}^2 \to \Re$ and think of it as a column vector on $\mathcal{N}^2$, whose (x_1, x_2)th element is $f(x_1, x_2)$. Now for an $\mathcal{N}^2 \times \mathcal{N}^2$ matrix (say Q), denote by Qf the column vector on $\mathcal{N}^2$, whose (x_1, x_2)th element (i.e., $(Qf)(x_1, x_2)$) is the product of the (x_1, x_2)th row of Q and the column vector f.

Define $\sigma : \mathcal{N}^2 \to \Re$, with $\sigma(x_1, x_2) = x_1 + x_2$. Then $(P_v\sigma)(y_1, y_2)$ is the expected total population in the two queues given that the queue lengths one step back were (y_1, y_2) and the previous control was v.

Let $\alpha_1(\underline{x}) = (x_1 + 1, x_2)$, $\alpha_2(\underline{x}) = (x_1, x_2 + 1)$, $\delta_1(\underline{x}) = ((x_1 - 1)^+, x_2)$, $\delta_2(\underline{x}) = (x_1, (x_2 - 1)^+)$, $\overline{\lambda} = 1 - \lambda$, $\overline{\mu} = 1 - \mu$. Defining $(P\sigma)(\underline{y}) = \mu\overline{\mu}\{\sigma(\delta_1(\underline{y})) + \sigma(\delta_2(\underline{y}))\} + \mu^2\sigma(\delta_1(\delta_2(\underline{y}))) + \overline{\mu}^2\sigma(\underline{y})$, we see that $(P_v\sigma)(\underline{y}) = \overline{\lambda}(P\sigma)(\underline{y}) + \lambda(P\sigma)(\alpha_v\underline{y})$.

d) One-step cost at the n-th step : $c(s(n), u(n)) = E[a(n) + q_1(n) + q_2(n) | s(n)] = \lambda + (P_{u(n-1)}\sigma)(q_1(n-1), q_2(n-1))$. $c(s(n), u(n))$ does not depend on $u(n)$, since the arrival at n, if any, must be routed to one of the two queues.

e) Performance criterion : The discounted cost criterion: $J^{\beta}(\pi, s(0)) = E^{\pi}_{s(0)}[\sum_{n=0}^{\infty} \beta^n c(s(n), u(n))]$ We note that this criterion is easily seen to be equivalent to our original performance criterion.

2.1 Optimality of JSEQ for one step delay

Define, for $d \in \{1,2\}$, $V_d^*(x_1, x_2) = \min_{\pi} E^{\pi}_{(x_1, x_2, d)}[\sum_{n=0}^{\infty} \beta^n (\lambda + q_1(n) + q_2(n))]$. V_d^* for $d \in \{1, 2\}$ are viewed as column vectors on $\mathcal{N}^2$. Then the Dynamic Programming Equations are:

$$V_1^*(x_1, x_2) = \lambda + (P_1\sigma)(x_1, x_2) + \beta \min\{(P_1 V_1^*(x_1, x_2), (P_1 V_2^*)(x_1, x_2)\} \quad (1)$$

$$V_2^*(x_1, x_2) = \lambda + (P_2\sigma)(x_1, x_2) + \beta \min\{(P_2 V_1^*)(x_1, x_2), (P_2 V_2^*)(x_1, x_2)\} \quad (2)$$

We have shown that the optimal value function $(V_1^*(x_1, x_2), V_2^*(x_1, x_2))$ has the following properties:

CH3229-2/92/0000-0997$1.00 © 1992 IEEE

P1) $V_i^*(\alpha_j \underline{x}) \geq V_i^*(\underline{x})$, $i,j \in \{1,2\}$, i.e., $V_1^*(.)$ and $V_2^*(.)$ are co-ordinatewise increasing,

P2) $V_1^*(\underline{x}) = V_2^*(\underline{x}')$, where $(x_1,x_2)' = (x_2,x_1)$,

P3) $\forall \underline{x}$ with $x_1 < x_2$, $V_1^*(\underline{x}) \leq V_2^*(\underline{x})$

P4) $\forall \underline{x}$ with $x_1 \leq x_2$, $V_i^*(\delta_1(\alpha_2(\underline{x}))) \geq V_j^*(\underline{x})$, $i,j \in \{1,2\}$.

Theorem 2.1.1 $\mathbf{V}^* = (V_1^*, V_2^*)$ *has properties P1 to P4.*

Proof: The proof relies on Lemma 8.4.2 of [3]: we omit the details for lack of space. See also [2],[5],[7]. □

Let, for $i = 1,2$, $s(n)=(\underline{x},i)$, $e_i(\underline{x})=x_i$. Define the Join the Shortest Expected Queue (JSEQ) policy as

$$u(n) = \begin{cases} 1 \;\; if & (P_i e_1)(\underline{x}) \;<\; (P_i e_2)(\underline{x}) \\ 2 \;\; otherwise & \end{cases} \tag{3}$$

because $E[q_1(n)|s(n)] = (P_i e_1)(\underline{x})$, and $E[q_2(n)|s(n)] = (P_i e_2)(\underline{x})$.

Theorem 2.1.2 *For $k = 1$, the JSEQ policy is optimal.*

Proof: See [4]. □

JSEQ can also be expressed as follows. Suppose $s(n) = (\underline{x}, i)$, $i \in \{1,2\}$. Then

$$u^*(n) = \begin{cases} 1 \;\; if \;\; (x_1 < x_2) \;\; or \;\; (x_1 = x_2, i = 2) \\ 2 \;\; if \;\; (x_1 > x_2) \;\; or \;\; (x_1 = x_2, i = 1) \end{cases} \tag{4}$$

We have also shown, via a counter-example, that JSEQ is not optimal for $k \geq 2$. See [4].

3 Optimal Customer Acceptance/Rejection at a Single Queue

In this section, we consider an extension of the classical problem of optimal acceptance/rejection of arrivals to a queue (see, for example, [6]), with the additional feature that the controller is permitted only to observe queue length information delayed by k steps. Proceeding formally, let $q(t)$ denote the queue length process. At time t, $t \in \{0,1,2,\ldots\}$, the controller computes a control value $u(t) \in \{0,1\}$, and is allowed only to observe the queue lengths till time $t-k$, and all controls till time $t-1$. In particular, at time 0, the controller knows $q(-k)$, and $\{u(-k)$, $u(-k+1)$, $\ldots$, $u(-1)\}$. The problem is for the controller to choose $\{u(0), u(1), \ldots\}$ to optimise a cost function.

Arrivals and departures occur as follows. An arrival occurs to the system with probability λ at $t = n+$, $n \geq -k$. The arrival is accepted if $u(n) = 1$, otherwise it is rejected. A departure occurs from a non-empty queue with probability μ at $t = n-$, $n \geq -k+1$.

We need a policy π for choosing $\{u(0), u(1), \ldots\}$ so as to minimise the following cost function: $E^{\pi}_{s(0)}[\sum_{n=0}^{\infty} \beta^n (bq(n) - \lambda(1-b)u(n))]$; $bq(n) - \lambda(1-b)u(n)$ is the expected cost in the interval n if the holding cost per customer per time step is b, and the reward for customer acceptance is 1.

As in Section 2, we formulate the problem as a CO-CMC for $k = 1$. The elements of the CO-CMC are as follows:

a) State at time n : $s(n) = (q(n-1), u(n-1))$, $n \in \mathcal{N}$. So the state space is $\mathcal{N} \times \{0,1\}$.

b) Action at time n : $u(n) \in \{0,1\}$. So the action space is $\{0,1\}$.

c) Transition probabilities : Let $y = (y_1, r)$, $l = (l_1, z)$, $y_1, l_1 \in \mathcal{N}$, $r, z \in \{0,1\}$. Then, $P(s(n+1) = l|s(n) = y, u(n) = \delta) = I\{z = \delta\} P(q(n) = l_1 | q(n-1) = y_1, u(n-1) = r)$.

We denote by P_r the $\mathcal{N} \times \mathcal{N}$ matrix with elements Prob.$(q(n) = l_1 | s(n) = (y_1, r))$. Define two column vectors on $\mathcal{N}$ as follows: $b_0 = b(0,1,2,3,\ldots)^t$, where $(\cdots)^t$ denotes transpose, and $b_1 = b_0 - \lambda(1-b)\underline{1}$, where $\underline{1}$ is the column vector of all 1's.

d) One step expected cost: let $s(n) = (x, r)$ and $u(n) = u \in \{0,1\}$; then using the same notation as in Section 2, we define $c((x,r),u) = (P_r b_0)(x) - \lambda(1-b)u$

e) Performance criterion : we use the discounted cost criterion: $J^{\beta}(\pi, s(0)) = E^{\pi}_{s(0)}[\sum_{n=0}^{\infty} \beta^n c(s(n), u(n))]$, which can be seen to be the same as that displayed earlier.

3.1 Optimality of a Two Threshold Policy for $k = 1$

Let the initial state be $s(0) = (x, r)$. Define

$$V_r^*(x) = \min_{\pi} E^{\pi}_{(x,r)}[\sum_{n=0}^{\infty} \beta^n c(s(n), u(n))]$$

The Dynamic Programming Equations (DPE) are :

$$V_0^*(x) = (P_0 b_0)(x) + \beta(P_0 V_0^*)(x) + \beta \min\{0, (P_0(V_1^* - V_0^*))(x) - \frac{\lambda(1-b)}{\beta}\} \tag{5}$$

$$V_1^*(x) = (P_1 b_0)(x) + \beta(P_1 V_0^*)(x) + \beta \min\{0, (P_1(V_1^* - V_0^*))(x) - \frac{\lambda(1-b)}{\beta}\} \tag{6}$$

We have shown that the optimal value function $(V_0^*(x), V_1^*(x))$ has the following properties:

P1) For $i = 0,1$ and $\forall x \in \mathcal{N}$, $V_i^*(\alpha(x)) \geq V_i^*(x)$, i.e., $V_0^*(.)$ and $V_1^*(.)$ are increasing with x

P2) $\forall x \in \mathcal{N}$, $[V_1^*(x) - V_0^*(x)] \geq 0$

P3) $\forall x \in \mathcal{N}$, $[V_1^*(\alpha(x)) - V_0^*(\alpha(x))] \geq [V_1^*(x) - V_0^*(x)]$, i.e., $(V_1^*(x) - V_0^*(x))$ is increasing with x

P4) $\forall x \in \mathcal{N}$, $[V_0^*(x+1) - V_1^*(x)] \geq 0$

P5) $\forall x \in \mathcal{N} - \{0\}$, $[V_0^*(\alpha(x)) - V_1^*(x)] \geq [V_0^*(x) - V_1^*(\delta(x))]$, i.e., $(V_0^*(x+1) - V_1^*(x))$ is increasing with x.

Theorem 3.1.1 $(V_0^*(x), V_1^*(x))$ *has P1 to P5.*

Proof: The proof relies on Lemma 8.4.2 of [3].□

Theorem 3.1.2 *For a delay of* 1 *slot, the optimal policy has the following* threshold *structure: there exist $m_0, m_1 \in \mathcal{N}$, $m_0 \geq m_1 > 0$ such that*

$$if\; u(n-1) = 0 \;then\; u(n) = \begin{cases} 1 \;\; if \;\; q(n-1) \;<\; m_0 \\ 0 \;\; if \;\; q(n-1) \;\geq\; m_0 \end{cases}$$

$$if\; u(n-1) = 1 \;then\; u(n) = \begin{cases} 1 \;\; if \;\; q(n-1) \;<\; m_1 \\ 0 \;\; if \;\; q(n-1) \;\geq\; m_1 \end{cases}$$

Further, $m_0 \geq m_1$.

Proof: This is immediate from Theorem 3.1.1, and the Dynamic Programming Equations 5 and 6. □

References

[1] Ephremides A, Varaiya P, Walrand J 1980 A simple dynamic routing problem. *IEEE Trans. Autom. Control. 25: 690-693*

[2] Hajek B 1984 Optimal control of two interacting service stations. *IEEE Trans. Autom. Control. 29: 491-499*

[3] Kumar P R, Varaiya P 1986 *Stochastic Systems: Estimation, Identification and Adaptive Control.* (Englewood Cliffs, NJ: Prentice Hall)

[4] Kuri J, Kumar Anurag 1992 Optimal Control of Arrivals to Queues with Delayed Queue Length Information. *Detailed manuscript.*

[5] Lin W, Kumar P R 1987 Optimal control of a queueing system with two heterogeneous servers. *IEEE Trans. Autom. Control. 29: 696-703*

[6] Stidham S 1985 Optimal control of admission to a queueing system. *IEEE Trans. Autom. Control. 30: 705-713*

[7] Walrand J 1988 *An Introduction to Queueing Networks.* (Englewood Cliffs, NJ: Prentice Hall)

[8] Weber R R 1978 On the optimal assignment of customers to parallel queues. *J. Appl. Probab. 15: 406-413*

[9] Winston W 1977 Optimality of the shortest line discipline. *J. Appl. Probab. 14: 181-189*