Chapter 7
Markovian Polling Systems: Functional Computation for Mean Waiting Times and its Computational Complexity

Tetsuji Hirayama

Abstract We consider Markovian polling systems in which a single server serves J stations with Poisson arrivals and general service times. After completing a service period at station i, the server selects station j with probability p_{ij} and visits the station after spending a switchover time. We use the functional computation for mean waiting times that has been investigated in our previous research on multiclass M/G/1 type systems (e.g., [1] and [2]), which is different from the buffer occupancy method used in [3]. The advantages of the functional computation method are (1) its wide applicability to the analysis of M/G/1 type multiclass queues, and (2) its rather small computational complexity compared with the buffer occupancy method.

7.1 Introduction

A polling system is a multiclass queueing system in which a single server serves customers arriving at J stations according to some scheduling algorithm. It has been receiving much attention because of its ability to model a large variety of systems including computer communication networks, intelligent production systems, and transportation systems (e.g., [4] and [5]).

Several methods of analyzing various polling systems have been investigated. The leading method is the buffer occupancy method (e.g., [6]–[8]). This method has been used to analyze not only the standard system models but also various variants of the models that include a system with a mixture of exhaustive and gated disciplines [9], a system with simultaneous arrivals [10], a system with customers' feedback [11], and a nondeterministic polling system [3], and so on.

T. Hirayama
Graduate School of Systems and Information Engineering, University of Tsukuba,
Ibaraki 305-8573, Japan
e-mail: hirayama@cs.tsukuba.ac.jp

W. Yue et al. (eds.), *Advances in Queueing Theory and Network Applications,*

The other methods have also been investigated (e.g., [2], [12]–[14]). A fundamental survey of the analysis of polling systems was given in [15], and a detailed explanatory survey of these methods was given in [4]. The descendant set technique [16] has taken another approach to obtain the moments of the buffer occupancy variables, and was used to analyze a state-dependent polling system [17]. A stochastic decomposition was used to obtain a pseudo-conservation law for a weighted sum of the mean waiting times under various scheduling algorithms [18]. Another type of a decomposition theorem that relates a system with nonzero switchover times to a system with zero switchover times was investigated in [19].

Many of the research efforts listed above were concerned with the cyclic systems. On the other hand, various polling schemes other than cyclic have been investigated. Random polling systems in which the server next visits station j stochastically with probability p_j were considered by Kleinrock and Levy [20]. They were used to analyze the distributed access scheme to communication channels [4], [20].

Srinivasan [3] extended their analysis to nondeterministic polling systems (including Markovian polling systems) in which the server moves among stations according to general stochastic rules. Markovian polling systems with single buffers were investigated by Chung, Un, and Jung [21]. A system in which the server visits stations according to an arbitrary polling sequence (or table) of stations was considered by Baker and Rubin [22]. In this system, stations can be given higher priority by being listed more frequently in the polling table. Boxma, Levy, and Weststrate [23] found (approximate) formulas and procedures for determining the visit frequencies that optimize the system performances.

In this chapter, we consider Markovian polling systems in which a single server serves customers at J stations, and obtain their mean waiting times. Customers arrive according to Poisson processes and their service time distributions are general. After completing a service period at station i, the server selects station j with probability p_{ij} and visits it after spending a switchover time. The customer selection rule at each station is either gated or exhaustive. Although the system was already solved by the buffer occupancy method in [3], we take the other method (functional computation) that has been investigated in our previous research on multiclass M/G/1 type systems (e.g., [1], [2], [24], and [25]).

The key skill of our functional computation method is to consider the expected waiting time of a customer conditioned on the system state at its arrival epoch and represent it as a function of the system state. The advantages of the method are

(1) Its wide applicability to the analysis of mean waiting times in M/G/1 type multiclass queues
(2) Its rather small computational complexity necessary to calculate the mean waiting times for all stations as compared with the buffer occupancy method

Our method was initially applied to multiclass M/G/1 queues with priority [24], and then extended to the systems with customers' feedback [1]. Polling systems were initially investigated by our method in [2], and their multiclass extensions with customers' feedback were investigated in [25]. In all of the models, we have obtained the linear functional expressions for the conditional expected waiting (or sojourn) times, which are the key property of our method, although the derivation procedures themselves are distinct among the models.

As for the computational complexity for computing the mean waiting times for all stations, our functional computation (for the Markovian polling system) requires us to solve $2J$ sets of $O(J)$ linear equations and a set of $O(J^2)$ (steady-state) linear equations. This means that our method at most requires $O(J^6)$ numerical operations. Furthermore a successive approximation method can be applied to solving the set of the latter steady-state linear equations, and then it can be shown that our method requires $O(J^4) + O(J^3 N)$ numerical operations where N is the number of its iterations. On the other hand, the buffer occupancy method requires us to solve the $O(J^3)$ linear equations for deriving the mean waiting times for all stations. If a successive approximation is applied to solving them, the method requires $O(J^4 N')$ numerical operations where N' is the number of its iterations.[1] Numerical examples are given in Sect. 7.6 of this chapter in order to compare the actual computational times in our method with those in the buffer occupancy method.

The rest of this chapter is organized as follows. In Sect. 7.2 we first define the system state that represents an evolution of the system. Its components include the numbers of customers and the remaining service time of a customer being served, and so on. Then we define some types of the expected waiting times for each customer conditioned on the system state at its arrival or relative polling instants. It is shown that these conditional expectations satisfy the "polling equation." In Sect. 7.3 we obtain the explicit expressions for some of the conditional expected waiting times. We further obtain the conditional expected numbers of customers at the next polling instants. In Sect. 7.4 the explicit expression for the overall expected waiting time is obtained by solving the polling equation. It can be shown that the expression has the linear functional form. In Sect. 7.5 the mean waiting times and the mean numbers of customers in a steady-state are obtained from the expression by using the generalized Little's formula and the PASTA property. Then we discuss the computational complexity of our functional computation method in detail in Sect. 7.6.

7.2 Model Description

In this section, we describe our model of the Markovian polling systems. A single server serves J groups of customers at J stations with infinite buffer capacities. Customers arrive at station i from outside the system according to a Poisson process with rate λ_i, and are called i-customers $(i = 1, \ldots, J)$. The overall arrival rate is denoted by $\lambda = \sum_{i=1}^{J} \lambda_i$. These customers are numbered in order of arrival, and let

[1] For a cyclic or random polling system, only $O(J^3)$ numerical operations are required for each iteration of the approximation for the buffer occupancy equations. But for a Markovian polling system, $O(J^4)$ operations are required for each iteration and the overall complexity becomes $O(J^4 N')$. For more detail, see [26] and (4.14) in [3].

c^e and τ_0^e denote the eth arriving customer itself and its arrival epoch, respectively $(e = 1, 2, \ldots)$.[2]

Service times S_i of i-customers are independently, identically, and arbitrarily distributed with mean $\mathrm{E}[S_i] > 0$ and second moment $\overline{s_i^2}$. Customers are served according to a predetermined scheduling algorithm defined below. The service is nonpreemptive. After receiving a service, each customer departs from the system. We define resource utilizations $\rho_i = \lambda_i \mathrm{E}[S_i]$, and put the usual assumption that $\rho = \sum_{i=1}^{J} \rho_i < 1$.

After completing a service period (defined below) at station i, the server selects a station in a Markovian manner where station j is selected with probability p_{ij}, and then visits station j after spending an arbitrarily distributed switchover time with mean $\overline{s_{ij}^o}$ and second moment $\overline{s_{ij}^{o2}}$, $(i, j = 1, \ldots, J)$. Let $\mathbf{P} = (p_{ij} : i, j = 1, \ldots, J)$ be the switching probability matrix, and assume that the Markov chain generated by the transition probability matrix $\mathbf{P}$ is irreducible. Furthermore, the arrival processes, the service times, and the server switching processes are assumed to be independent of each other.

The system is separated into two parts which are called the "service facility" and the "waiting room." There is a gate at each station between its queue in the waiting room and its queue in the service facility. And each arriving customer enters the queue in the service facility when the gate is opened; otherwise, it enters the queue in the waiting room. When the server visits a station, its gate is opened in order to admit some customers at the station to the service facility. The server serves the customers in the service facility until the server empties it, and then visits another station. Because the gates of the stations that are not visited by the server are closed, all customers at such stations must wait for service in the waiting room.

Each time interval from when the server visits a station until the first time when the server empties the service facility is called a *service period*.[3] Each time interval when the server switches from a station to another station is called a *switchover period*. Let $\Pi = \{1, \ldots, J\}$ be the set of (indices of) the service periods where $i \in \Pi$ denotes the service period of station i. And let $\Pi^s = \{(i, j) : i, j = 1, \ldots, J\}$ be the set of (indices of) the switchover periods where (i, j) denotes a switchover period from station i to station j.

A scheduling algorithm is specified as follows: (1) Selection order of the stations by the server, which is the Markovian as described before, (2) customer selection rule at each station used when the server admits customers into the service facility, which is either gated or exhaustive, and (3) service order of customers in the service facility, which is First-Come First-Served (FCFS).

When the server selects one of the stations with the gated rule, all customers staying at the station just when the server visits it enter its queue in the service facility, and then the gate is immediately closed. $\mathscr{H}_g$ denotes the set of stations with the gated rule. When the server selects one of the stations with the exhaustive rule,

[2] These customers arrive from outside the system according to a Poisson process with rate λ, and each of them becomes an i-customer with probability λ_i/λ when it arrives $(i = 1, \ldots, J)$.

[3] A time epoch when the server visits a station is called a service period beginning epoch or a polling instant.

the gate of the station remains open (i.e., customers arriving at the station later may still enter the service facility) and the server continues to serve all customers until the station is cleared of customers for the first time. The service period of the station finishes at this time, and its gate is closed. $\mathscr{H}_e$ denotes the set of the stations with the exhaustive rule.

Let us consider the system operating under a specified scheduling algorithm. For any e $(e = 1,2,\ldots)$, let $\{\tau_k^e : k = 1,2,\ldots\}$ be a sequence of all polling instants (i.e., service period beginning epochs) of all stations that occur after the c^es' arrival epoch.[4] Furthermore let $X_S^e(t)$ denote the station at which c^e stays at time t, or $X_S^e(t) = 0$ if it does not stay in the system at time t. Let $\mathscr{R}, \mathscr{R}_+, \mathscr{I}_+$ be, respectively, the set of real numbers, the set of nonnegative real numbers, and the set of nonnegative integers. For any event $\mathscr{K}$, let

$$\mathbf{1}\{\mathscr{K}\} = \begin{cases} 1, & \text{if event } \mathscr{K} \text{ is true} \\ 0, & \text{if event } \mathscr{K} \text{ is false.} \end{cases}$$

Then let $\kappa(t)$ denote a period that the system experiences at time t; that is the server is in a service period of station $\kappa(t)$ if $\kappa(t) \in \Pi$, or the server is in a switchover period from station i to station j if $\kappa(t) = (i,j) \in \Pi^s$. Let $r(t)$ denote the remaining service time of a customer being served at time t if $\kappa(t) \in \Pi$, or the remaining length of a switchover period if $\kappa(t) \in \Pi^s$.

The number of i-customers in the service facility at time t (who are not being served) is denoted by $g_i(t)$, and the number of i-customers in the waiting room at time t is denoted by $n_i(t)$. Let $\mathbf{g}(t) = (g_1(t),\ldots,g_J(t)) \in \mathscr{I}_+^J$, and let $\mathbf{n}(t) = (n_1(t),\ldots,n_J(t)) \in \mathscr{I}_+^J$. We also specify the other information $L(t)$ of the system at time t. The sample paths of these processes are assumed to be left-continuous with right-hand limits, except for $X_S^e(t), \kappa(t)$, and $L(t)$ which are right-continuous with left-hand limits.

Let us consider transition epochs of these processes consisting of customer arrival epochs, service completion epochs, and switchover period completion epochs. Then we define the stochastic process as

$$\mathscr{Q} = \{\mathbf{Y}(t) = (\kappa(t), r(t), \mathbf{g}(t), \mathbf{n}(t), L(t)) : t \geq 0\} \tag{7.1}$$

which represents an evolution of the system. For any scheduling algorithm defined above, $\mathscr{Q}$ may embed a Markov process. Possible values of $\mathbf{Y}(t)$ $(t \geq 0)$ are called *states*, and the state space of $\mathscr{Q}$ is denoted by $\mathscr{E}$.

We define three types of the performance measures of customer c^e $(e = 1,2,\ldots)$. The first type is related to the c^es' waiting times in the waiting room. We define for any $t \geq 0$ and $i = 1,\ldots,J$,

$$C_{Wi}^e(t) = \begin{cases} 1, & \text{if } c^e \text{ stays in the waiting room as an } i\text{-customer at time } t \\ 0, & \text{otherwise.} \end{cases} \tag{7.2}$$

[4] Note that τ_0^e is the customer's arrival epoch, and we assume that $\tau_0^e < \tau_1^e < \tau_2^e < \cdots$.

The c^es' waiting time spent in the waiting rooms is defined by

$$W_i^e = \int_0^\infty C_{Wi}^e(t)dt, \qquad (i = 1,\ldots,J). \tag{7.3}$$

Then, for $l = 0,1,2\ldots$, the expected waiting times in the waiting room during the time interval $[\tau_l^e, \tau_{l+1}^e)$ conditioned on the state of the system are defined by

$$W_i^0(\mathbf{Y},e,l) = \mathrm{E}\left[\int_{\tau_l^e}^{\tau_{l+1}^e} C_{Wi}^e(t)dt \,\middle|\, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = i\right] \tag{7.4}$$

for $\mathbf{Y} \in \mathscr{E}$, $i = 1,\ldots,J$.

The second type of the performance measures is related to the pieces of the c^es' waiting times in the waiting room. Let

$$H_i^e(k) = \int_0^\infty C_{Wi}^e(t)\mathbf{1}\{\kappa(t) = k\}dt, \qquad (i = 1,\ldots,J,\ k \in \Pi \cup \Pi^s). \tag{7.5}$$

$H_i^e(k)$ denotes the c^es' waiting times in the waiting room spent while the system is in period k. For $l = 0,1,2,\ldots$, the expected waiting times after time τ_l^e conditioned on the state of the system are defined by

$$H_i(\mathbf{Y},e,l,k) = \mathrm{E}\left[\int_{\tau_l^e}^{\infty} C_{Wi}^e(t)\mathbf{1}\{\kappa(t) = k\}dt \,\middle|\, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = i\right], \tag{7.6}$$

$$H_i^0(\mathbf{Y},e,l,k) = \mathrm{E}\left[\int_{\tau_l^e}^{\tau_{l+1}^e} C_{Wi}^e(t)\mathbf{1}\{\kappa(t) = k\}dt \,\middle|\, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = i\right] \tag{7.7}$$

for $i = 1,\ldots,J$, $k \in \Pi \cup \Pi^s$, $\mathbf{Y} \in \mathscr{E}$. Then the following "polling equation" holds.

$$H_i(\mathbf{Y},e,l,k) = \begin{cases} H_i^0(\mathbf{Y},e,l,k) + \mathrm{E}[H_i(\mathbf{Y}(\tau_{l+1}^e),e,l+1,k)|\mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = i], \\ \qquad \text{if } (\kappa_0 \neq i) \text{ or } (\kappa_0 = i \in \mathscr{H}_g, l = 0) \\ 0, \quad \text{if } (\kappa_0 = i \in \mathscr{H}_e) \text{ or } (\kappa_0 = i \in \mathscr{H}_g, l > 0) \end{cases} \tag{7.8}$$

for $\mathbf{Y} = (\kappa_0, r, \mathbf{g}, \mathbf{n}, L) \in \mathscr{E}$, $i = 1,\ldots,J$, $l = 0,1,\ldots$, $k \in \Pi \cup \Pi^s$.

The third type of the performance measures is related to the c^es' waiting times in the service facility. We define for any $t \geq 0$ and $i = 1,\ldots,J$,

$$C_{Fi}^e(t) = \begin{cases} 1, \text{ if } c^e \text{ is in the service facility as an } i\text{-customer and} \\ \quad \text{is not served at time } t \\ 0, \text{ otherwise.} \end{cases} \tag{7.9}$$

The c^es' waiting time in the service facility is defined by

$$F_i^e = \int_0^\infty C_{Fi}^e(t)dt, \qquad (i = 1,\ldots,J). \tag{7.10}$$

The expected waiting times in the service facility after time τ_0^e conditioned on the state of the system are defined by

$$F_i(\mathbf{Y},e) = \mathrm{E}\left[\int_{\tau_0^e}^{\infty} C_{Fi}^e(t)dt \,\middle|\, \mathbf{Y}(\tau_0^e) = \mathbf{Y}, X_S^e(\tau_0^e) = i\right] \tag{7.11}$$

for $\mathbf{Y} \in \mathscr{E}$, $i = 1,\dots,J$.

7.3 Expressions for $W_j^0(\cdot), H_j^0(\cdot), F_j(\cdot)$, and Related Quantities

In this section we obtain the conditional expected waiting times $W_j^0(\cdot), H_j^0(\cdot)$, and $F_j(\cdot)$ of a j-customer ($j = 1,\dots,J$). We also consider the expected number of customers at the next polling instant. We observe a specific customer c^e assuming that it is a j-customer ($e = 1,2,\dots$).

7.3.1 *Expressions for $W_j^0(\cdot), H_j^0(\cdot)$, and $F_j(\cdot)$*

Let $l = 0,1,2,\dots$ and let $\mathbf{Y} = (\kappa_0, r, \mathbf{g}, \mathbf{n}, L) \in \mathscr{E}$ be the system state at time τ_l^e where $\mathbf{g} = (g_1,\dots,g_J)$ and $\mathbf{n} = (n_1,\dots,n_J)$. Because we assume that c^e is a j-customer, $X_S^e(\tau_l^e) = j$. When we consider the c^es' expected waiting time in the waiting room $W_j^0(\mathbf{Y},e,l)$ during the time interval $[\tau_l^e, \tau_{l+1}^e)$, we consider the following cases according to $\kappa_0 = \kappa(\tau_l^e)$, which is the period at time τ_l^e. For $\kappa_0 \in \mathscr{H}_g$, we have

$$W_j^0(\mathbf{Y},e,l) = \begin{cases} n_{\kappa_0}\mathrm{E}[S_{\kappa_0}] + \sum_{\kappa_1=1}^{J} p_{\kappa_0\kappa_1}\overline{s_{\kappa_0\kappa_1}^o}, & \kappa_0 \neq j,\ (l > 0) \\ 0, & \kappa_0 = j,\ (l > 0) \\ r + g_{\kappa_0}\mathrm{E}[S_{\kappa_0}] + \sum_{\kappa_1=1}^{J} p_{\kappa_0\kappa_1}\overline{s_{\kappa_0\kappa_1}^o}, & (l = 0). \end{cases} \tag{7.12}$$

For $\kappa_0 \in \mathscr{H}_e$, we have

$$\begin{aligned} &W_j^0(\mathbf{Y},e,l) \\ &= \begin{cases} (n_{\kappa_0}\mathrm{E}[S_{\kappa_0}])/(1-\rho_{\kappa_0}) + \sum_{\kappa_1=1}^{J} p_{\kappa_0\kappa_1}\overline{s_{\kappa_0\kappa_1}^o}, & \kappa_0 \neq j,\ (l > 0) \\ (r + g_{\kappa_0}\mathrm{E}[S_{\kappa_0}])/(1-\rho_{\kappa_0}) + \sum_{\kappa_1=1}^{J} p_{\kappa_0\kappa_1}\overline{s_{\kappa_0\kappa_1}^o}, & \kappa_0 \neq j,\ (l = 0) \\ 0, & \kappa_0 = j,\ (l \geq 0). \end{cases} \end{aligned} \tag{7.13}$$

For $\kappa_0 \in \Pi^s$, we have

$$W_j^0(\mathbf{Y},e,l) = \begin{cases} 0, & (l > 0) \\ r, & (l = 0). \end{cases} \tag{7.14}$$

Because $H_j^0(\mathbf{Y},e,l,k)$ is a piece of the expected waiting time $W_j^0(\mathbf{Y},e,l)$, it is given by appropriately choosing the parts of $W_j^0(\cdot)$. For $\kappa_0 \in \mathscr{H}_g$, we have

$$H_j^0(\mathbf{Y},e,l,k) = \begin{cases} n_{\kappa_0}\mathrm{E}[S_{\kappa_0}], & k=\kappa_0,\ (\kappa_0 \neq j,\ l>0) \\ p_{\kappa_0\kappa_1}\overline{s^o_{\kappa_0\kappa_1}}, & k=(\kappa_0,\kappa_1)\in\Pi^s,\ (\kappa_0\neq j,\ l>0) \\ 0, & (\kappa_0=j,\ l>0) \\ r+g_{\kappa_0}\mathrm{E}[S_{\kappa_0}], & k=\kappa_0,\ (l=0) \\ p_{\kappa_0\kappa_1}\overline{s^o_{\kappa_0\kappa_1}}, & k=(\kappa_0,\kappa_1)\in\Pi^s,\ (l=0) \\ 0, & \text{otherwise,} \end{cases} \tag{7.15}$$

where $\kappa_1 = 1,\ldots,J$. And for $\kappa_0 \in \mathscr{H}_e$, we have

$$H_j^0(\mathbf{Y},e,l,k) = \begin{cases} (n_{\kappa_0}\mathrm{E}[S_{\kappa_0}])/(1-\rho_{\kappa_0}), & k=\kappa_0,\ (\kappa_0 \neq j,\ l>0) \\ p_{\kappa_0\kappa_1}\overline{s^o_{\kappa_0\kappa_1}}, & k=(\kappa_0,\kappa_1)\in\Pi^s,\ (\kappa_0\neq j,\ l>0) \\ (r+g_{\kappa_0}\mathrm{E}[S_{\kappa_0}])/(1-\rho_{\kappa_0}), & k=\kappa_0,\ (\kappa_0\neq j,\ l=0) \\ p_{\kappa_0\kappa_1}\overline{s^o_{\kappa_0\kappa_1}}, & k=(\kappa_0,\kappa_1)\in\Pi^s,\ (\kappa_0\neq j,\ l=0) \\ 0, & (\kappa_0=j,\ l\geq 0) \\ 0, & \text{otherwise,} \end{cases} \tag{7.16}$$

where $\kappa_1 = 1,\ldots,J$. For $\kappa_0 \in \Pi^s$, we have

$$H_j^0(\mathbf{Y},e,l,k) = \begin{cases} 0, & (l>0) \\ r, & k=\kappa_0,\ (l=0) \\ 0, & \text{otherwise, } (l=0). \end{cases} \tag{7.17}$$

Because $F_j(\mathbf{Y},e)$ is the expected waiting time in the service facility, it is equal to the expected (remaining) service times of customers at station j at the c^es' arrival epoch τ_0^e. Then we have

$$F_j(\mathbf{Y},e) = \begin{cases} n_j\mathrm{E}[S_j], & j\in\mathscr{H}_g \\ n_j\mathrm{E}[S_j], & j\in\mathscr{H}_e \text{ and } j\neq\kappa_0 \\ r+g_j\mathrm{E}[S_j], & j\in\mathscr{H}_e \text{ and } j=\kappa_0. \end{cases} \tag{7.18}$$

7.3.2 System State at the Next Polling Instant

Let $l = 0, 1, 2, \ldots$ and let $\mathbf{Y} = (\kappa_0, r, \mathbf{g}, \mathbf{n}, L) \in \mathscr{E}$ be the system state at time τ_l^e where $\mathbf{g} = (g_1, \ldots, g_J)$ and $\mathbf{n} = (n_1, \ldots, n_J)$. We consider the system state at the next polling instant τ_{l+1}^e.

When we consider the system state (especially, the numbers of customers) at the next polling instant, we consider the following cases according to $\kappa_0 = \kappa(\tau_l^e)$, which is the period at time τ_l^e. For $\kappa_0 \in \mathscr{H}_g$, we can show that

$$\begin{aligned}
&\mathrm{E}[n_m(\tau_{l+1}^e) | \kappa(\tau_{l+1}^e) = \kappa_1, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j] \\
&= \begin{cases} n_m + \lambda_m \{n_{\kappa_0} \mathrm{E}[S_{\kappa_0}] + \overline{s_{\kappa_0\kappa_1}^o}\}, & m \neq \kappa_0,\ (l > 0) \\ \lambda_{\kappa_0} \{n_{\kappa_0} \mathrm{E}[S_{\kappa_0}] + \overline{s_{\kappa_0\kappa_1}^o}\}, & m = \kappa_0,\ (l > 0) \\ n_m + \mathbf{1}_{mj} + \lambda_m \{r + g_{\kappa_0} \mathrm{E}[S_{\kappa_0}] + \overline{s_{\kappa_0\kappa_1}^o}\}, & (l = 0) \end{cases}
\end{aligned} \tag{7.19}$$

for any $m, j, \kappa_1 \in \Pi$, where $\mathbf{1}_{mj} = \mathbf{1}\{m = j\}$. For $\kappa_0 \in \mathscr{H}_e$, we have

$$\begin{aligned}
&\mathrm{E}[n_m(\tau_{l+1}^e) | \kappa(\tau_{l+1}^e) = \kappa_1, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j] \\
&= \begin{cases} n_m + \lambda_m \{(n_{\kappa_0} \mathrm{E}[S_{\kappa_0}])/(1 - \rho_{\kappa_0}) + \overline{s_{\kappa_0\kappa_1}^o}\}, & m \neq \kappa_0,\ (l > 0) \\ n_m + \mathbf{1}_{mj} + \lambda_m \{(r + (g_{\kappa_0} + \mathbf{1}_{\kappa_0 j}) \mathrm{E}[S_{\kappa_0}])/(1 - \rho_{\kappa_0}) + \overline{s_{\kappa_0\kappa_1}^o}\}, & \\ & m \neq \kappa_0,\ (l = 0) \\ \lambda_{\kappa_0} \overline{s_{\kappa_0\kappa_1}^o}, & m = \kappa_0,\ (l \geq 0) \end{cases}
\end{aligned} \tag{7.20}$$

for any $m, j, \kappa_1 \in \Pi$. For $\kappa_0 \in \Pi^s$, we have

$$\begin{aligned}
&\mathrm{E}[n_m(\tau_{l+1}^e) | \kappa(\tau_{l+1}^e) = \kappa_1, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j] \\
&= \begin{cases} 0, & (l > 0) \\ n_m + \mathbf{1}_{mj} + \lambda_m r, & (l = 0). \end{cases}
\end{aligned} \tag{7.21}$$

Furthermore for any $m, j, \kappa_1 \in \Pi$, we obviously have

$$\mathrm{E}[g_m(\tau_{l+1}^e) | \kappa(\tau_{l+1}^e) = \kappa_1, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j] = 0. \tag{7.22}$$

7.3.3 Unified Forms: Linear Functional Expressions

From the analysis in this section, we can easily see the following important properties.

- The component $(\kappa_0, r, \mathbf{g}, \mathbf{n})$ of state $\mathbf{Y} = (\kappa_0, r, \mathbf{g}, \mathbf{n}, L) \in \mathscr{E}$ at epoch τ_l^e is sufficient to derive $W_i^0(\mathbf{Y}, e, l), H_j^0(\mathbf{Y}, e, l, k), F_j(\mathbf{Y}, e)$, and the conditional expected numbers of customers at time τ_{l+1}^e.
- These quantities are linear with respect to r and $(\mathbf{g}, \mathbf{n}) = (g_1, \ldots, g_J, n_1, \ldots, n_J)$.

For convenience, let $\mathbf{e}_j = (0,\ldots,0,\underbrace{1}_{j^{th}\text{ place}},0,\ldots,0) \in \mathscr{R}^{1\times J}$, and let $p_k = p_{\kappa_0,\kappa_1}$ for $k = (\kappa_0,\kappa_1) \in \Pi^s$. Then we have the following.

Proposition 7.1. *Let* $\mathbf{Y} = (\kappa_0, r, \mathbf{g}, \mathbf{n}, L) \in \mathscr{E}$, $j = 1,\ldots,J$, $e = 1,2,\ldots$, $l = 0,1,2,\ldots$ *and* $k \in \Pi \cup \Pi^s$. *Then we have*

$$H_j^0(\mathbf{Y},e,l,k) = \begin{cases} (\mathbf{g},\mathbf{n})\mathbf{h}_{10}^0(\kappa_0,j,k), & \kappa_0 \in \Pi,\ l>0,\ k\in\Pi \\ p_k h_{11}^0(\kappa_0,j,k), & \kappa_0 \in \Pi,\ l>0,\ k\in\Pi^s \\ r\varphi^0(\kappa_0,j,k) + (\mathbf{g},\mathbf{n})\mathbf{h}_{00}^0(\kappa_0,j,k), & \kappa_0 \in \Pi,\ l=0,\ k\in\Pi \\ p_k h_{01}^0(\kappa_0,j,k), & \kappa_0 \in \Pi,\ l=0,\ k\in\Pi^s \\ 0, & \kappa_0 \in \Pi^s,\ l>0,\ k\in\Pi\cup\Pi^s \\ 0, & \kappa_0 \in \Pi^s,\ l=0,\ k\in\Pi \\ r\varphi^0(\kappa_0,j,k), & \kappa_0 \in \Pi^s,\ l=0,\ k\in\Pi^s, \end{cases} \tag{7.23}$$

$$F_j(\mathbf{Y},e) = r\psi(\kappa_0,j) + (\mathbf{g},\mathbf{n})\mathbf{f}(\kappa_0,j), \tag{7.24}$$

where the above coefficients

$$\mathbf{h}_{a0}^0(\kappa_0,j,k) \in \mathscr{R}^{2J\times 1}, \qquad h_{a1}^0(\kappa_0,j,k) \in \mathscr{R}, \qquad (a=0,1),$$
$$\varphi^0(\kappa_0,j,k) \in \mathscr{R}, \qquad \psi(\kappa_0,j) \in \mathscr{R}, \qquad \mathbf{f}(\kappa_0,j) \in \mathscr{R}^{2J\times 1}$$

can be determined from the given system parameters through the expressions obtained in this section. Furthermore we have

$$\mathrm{E}[(\mathbf{g}(\tau_{l+1}^e),\mathbf{n}(\tau_{l+1}^e))|\kappa(\tau_{l+1}^e) = \kappa_1, \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j]$$
$$= \begin{cases} (\mathbf{g},\mathbf{n})\mathbf{U}_1(\kappa_0) + \mathbf{u}_1(\kappa_0,\kappa_1), & \kappa_0 \in \Pi,\ l>0 \\ r\upsilon(\kappa_0) + (\mathbf{g},\mathbf{n})\mathbf{U}_0(\kappa_0) + \mathbf{u}_0(j,\kappa_0,\kappa_1), & \kappa_0 \in \Pi,\ l=0 \\ \mathbf{0}, & \kappa_0 \in \Pi^s,\ l>0 \\ r\upsilon + (\mathbf{g},\mathbf{n})\mathbf{U}_0 + (\mathbf{0},\mathbf{e}_j), & \kappa_0 \in \Pi^s,\ l=0 \end{cases} \tag{7.25}$$

for $\kappa_1 \in \Pi$. *The above coefficients*

$$\mathbf{U}_1(\kappa_0) \in \mathscr{R}^{2J\times 2J}, \qquad \mathbf{u}_1(\kappa_0,\kappa_1) \in \mathscr{R}^{1\times 2J}, \qquad \upsilon(\kappa_0) \in \mathscr{R}^{1\times 2J},$$
$$\mathbf{U}_0(\kappa_0) \in \mathscr{R}^{2J\times 2J}, \qquad \mathbf{u}_0(j,\kappa_0,\kappa_1) \in \mathscr{R}^{1\times 2J}, \qquad \upsilon \in \mathscr{R}^{1\times 2J}, \qquad \mathbf{U}_0 \in \mathscr{R}^{2J\times 2J}$$

can be determined from the given system parameters through the expressions obtained in this section.

Note 1. We can simplify the expression for $H_j^0(\cdot)$ as follows:

$$H_j^0(\mathbf{Y},e,l,k) = \begin{cases} (\mathbf{g},\mathbf{n})\mathbf{h}_{10}^0(\kappa_0,j,k)+p_k h_{11}^0(\kappa_0,j,k), & l>0 \\ r\varphi^0(\kappa_0,j,k)+(\mathbf{g},\mathbf{n})\mathbf{h}_{00}^0(\kappa_0,j,k)+p_k h_{01}^0(\kappa_0,j,k), & l=0. \end{cases}$$

Because this expression introduces much labor into the numerical calculation, we adopt the above somewhat complicated expression. A similar result holds for the expression in Equation (7.25). □

Note 2. It should be noted from Equations (7.15) and (7.16) that

$$\begin{aligned} &H_j^0(\mathbf{Y},e,l,k)=0, \\ &\qquad (j\in\Pi,\ \mathbf{Y}=(\kappa_0,r,\mathbf{g},\mathbf{n},L)\in\mathscr{E},\ e=1,2,\ldots,\ l\geq 0,\ k\in\Pi\cup\Pi^s) \end{aligned}$$

if ($\kappa_0=j\in\mathscr{H}_e$) or ($\kappa_0=j\in\mathscr{H}_g$ and $l>0$). □

7.4 The Linear Functional Expression

In this section we obtain the expression for the performance measure $H_j(\cdot)$ by solving the polling equation. It can be shown that it has the linear functional form.

We define constants $\mathbf{h}_{10}(\kappa_0,j,k)\in\mathscr{R}^{2J\times 1}$ and $h_{11}(\kappa_0,j,k)\in\mathscr{R}$ that satisfy the following equations:

$$\mathbf{h}_{10}(\kappa_0,j,k) = \begin{cases} \mathbf{h}_{10}^0(\kappa_0,j,k)+\mathbf{U}_1(\kappa_0)\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}} p_{\kappa_0\kappa_1}\mathbf{h}_{10}(\kappa_1,j,k), \\ \qquad\qquad \kappa_0\neq j,\ \kappa_0\in\Pi,\ k\in\Pi \\ \mathbf{0}, \qquad \text{otherwise}, \end{cases} \tag{7.26}$$

$$h_{11}(\kappa_0,j,k) = \begin{cases} \displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}} p_{\kappa_0\kappa_1}\mathbf{u}_1(\kappa_0,\kappa_1)\mathbf{h}_{10}(\kappa_1,j,k) \\ \qquad +\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}} p_{\kappa_0\kappa_1}h_{11}(\kappa_1,j,k), \\ \qquad\qquad \kappa_0\neq j,\ \kappa_0\in\Pi,\ k\in\Pi \\ p_k h_{11}^0(\kappa_0,j,k)+\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}} p_{\kappa_0\kappa_1}h_{11}(\kappa_1,j,k), \\ \qquad\qquad \kappa_0\neq j,\ \kappa_0\in\Pi,\ k\in\Pi^s \\ 0, \qquad \text{otherwise} \end{cases} \tag{7.27}$$

for $j\in\Pi$. Furthermore for $k\in\Pi$, $\kappa_0\in\Pi\cup\Pi^s$, and $j\in\Pi$, let[5]

[5] Case 1: ($\kappa_0\neq j$ or $j\in\mathscr{H}_g$) and ($\kappa_0\in\Pi$); Case 2: $\kappa_0=j\in\mathscr{H}_e$; Case 3: $\kappa_0=(k_0,k_1)\in\Pi^s$.

$$\varphi(\kappa_0,j,k)=\begin{cases}\varphi^0(\kappa_0,j,k)+\upsilon(\kappa_0)\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}}p_{\kappa_0\kappa_1}\mathbf{h}_{10}(\kappa_1,j,k), & \text{case 1}\\ 0, & \text{case 2}\\ \upsilon\mathbf{h}_{10}(k_1,j,k), & \text{case 3,}\end{cases}$$

$$\mathbf{h}_{00}(\kappa_0,j,k)=\begin{cases}\mathbf{h}^0_{00}(\kappa_0,j,k)+\mathbf{U}_0(\kappa_0)\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}}p_{\kappa_0\kappa_1}\mathbf{h}_{10}(\kappa_1,j,k), & \text{case 1}\\ \mathbf{0}, & \text{case 2}\\ \mathbf{U}_0\mathbf{h}_{10}(k_1,j,k), & \text{case 3,}\end{cases}$$

$$h_{01}(\kappa_0,j,k)$$
$$=\begin{cases}\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}}p_{\kappa_0\kappa_1}\{\mathbf{u}_0(j,\kappa_0,\kappa_1)\mathbf{h}_{10}(\kappa_1,j,k)+h_{11}(\kappa_1,j,k)\}, & \text{case 1}\\ 0, & \text{case 2}\\ (\mathbf{0},\mathbf{e}_j)\mathbf{h}_{10}(k_1,j,k)+h_{11}(k_1,j,k), & \text{case 3.}\end{cases}$$

And for $k\in\Pi^s$, $\kappa_0\in\Pi\cup\Pi^s$ and $j\in\Pi$, let

$$\varphi(\kappa_0,j,k)=\begin{cases}0, & \kappa_0\in\Pi\\ \varphi^0(\kappa_0,j,k), & \kappa_0\in\Pi^s,\end{cases}$$

$$\mathbf{h}_{00}(\kappa_0,j,k)=\mathbf{0},$$

$$h_{01}(\kappa_0,j,k)=\begin{cases}p_k h^0_{01}(\kappa_0,j,k)+\displaystyle\sum_{\kappa_1\in\Pi\setminus\{j\}}p_{\kappa_0\kappa_1}h_{11}(\kappa_1,j,k), & \text{case 1}\\ 0, & \text{case 2}\\ h_{11}(k_1,j,k), & \text{case 3.}\end{cases}$$

Now we define the following function, and show that it gives the linear functional expression for the performance measure $H_j(\cdot)$ defined by (7.6).

Definition 7.1. The linear function is defined by

$$\hat{H}_j(\mathbf{Y},e,l,k)$$
$$=\begin{cases}r\varphi(\kappa_0,j,k)+(\mathbf{g},\mathbf{n})\mathbf{h}_{00}(\kappa_0,j,k)+h_{01}(\kappa_0,j,k), & l=0,\ k\in\Pi\\ r\varphi(\kappa_0,j,k)+h_{01}(\kappa_0,j,k), & l=0,\ k\in\Pi^s\\ (\mathbf{g},\mathbf{n})\mathbf{h}_{10}(\kappa_0,j,k)+h_{11}(\kappa_0,j,k), & l>0,\ k\in\Pi\\ h_{11}(\kappa_0,j,k), & l>0,\ k\in\Pi^s\end{cases}\tag{7.28}$$

for any $j\in\Pi$; $\mathbf{Y}=(\kappa_0,r,\mathbf{g},\mathbf{n},L)\in\mathscr{E}$; $e=1,2,\ldots$; $l=0,1,2,\ldots$ and $k\in\Pi\cup\Pi^s$.

Proposition 7.2. *The function $\hat{H}_{\cdot}(\cdot,\cdot,\cdot,k)$ ($k\in\Pi\cup\Pi^s$) defined by (7.28) satisfies the "polling equation" (7.8).*

Proof. See the appendix. □

Proposition 7.3. *The solution of the "polling equation" (7.8) is unique and hence*

$$H_j(\mathbf{Y},e,l,k) = \hat{H}_j(\mathbf{Y},e,l,k),$$
$$(j \in \Pi;\ \mathbf{Y} \in \mathscr{E};\ e = 1,2,\ldots;\ l = 0,1,2,\ldots;\ k \in \Pi \cup \Pi^s).$$

Proof. Because the proof of this proposition is similar to the proof of uniqueness of the solution for the feedback equation given in [1], it is omitted. □

7.5 Steady-State Values

We would like to obtain the steady-state values of the performance measures. We define the mean waiting time of j-customers[6] as follows:

$$\bar{w}_j = \lim_{N\to\infty} \frac{1}{N} \sum_{e=1}^{N} \mathrm{E}[W_j^e + F_j^e | X_S^e(\tau_0^e) = j], \qquad j = 1,\ldots,J. \tag{7.29}$$

In order to obtain the quantity, we define the following interim quantities:

$$\bar{H}_j(\kappa_0,k) = \lim_{N\to\infty} \frac{1}{N} \sum_{e=1}^{N} \mathrm{E}[H_j^e(k)\mathbf{1}\{\kappa(\tau_0^e) = \kappa_0\} | X_S^e(\tau_0^e) = j], \tag{7.30}$$

$$\bar{F}_j(\kappa_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{e=1}^{N} \mathrm{E}[F_j^e \mathbf{1}\{\kappa(\tau_0^e) = \kappa_0\} | X_S^e(\tau_0^e) = j] \tag{7.31}$$

for $j \in \Pi$ and $\kappa_0, k \in \Pi \cup \Pi^s$. The time average values of the system state are defined by

$$\tilde{\mathbf{Y}}^k = (k\tilde{q}^k, \tilde{r}^k, \tilde{\mathbf{g}}^k, \tilde{\mathbf{n}}^k, \tilde{L}^k) = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathrm{E}[\mathbf{Y}(s)\mathbf{1}\{\kappa(s) = k\}]ds \tag{7.32}$$

for $k \in \Pi \cup \Pi^s$ where $\tilde{\mathbf{g}}^k = (\tilde{g}_1^k,\ldots,\tilde{g}_J^k), \tilde{\mathbf{n}}^k = (\tilde{n}_1^k,\ldots,\tilde{n}_J^k)$.

For $k \in \Pi$, the steady-state value $\tilde{q}^k$, which is the long-run fraction of time that the system is in period k, is calculated as

$$\tilde{q}^k = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathrm{E}[\mathbf{1}\{\kappa(s) = k\}]ds = \lambda_k \mathrm{E}[S_k]. \tag{7.33}$$

For $k \in \Pi^s$, the steady-state value $\tilde{q}^k$ can be obtained in the following manner. Let π_i be the steady-state probability that the server selects station i at a polling instant. It can be easily shown that $\pi = (\pi_1,\ldots,\pi_J)$ is the steady-state probability of the Markov chain with the transition probability matrix $\mathbf{P}$. We can obtain it by solving $\pi\mathbf{P} = \pi$ and $\pi\mathbf{1} = 1$. Then the long-run fraction of time that the server is moving from station i to station j given that the system is in a switchover period is given by

[6] The time average values and the customer average values defined in this section are assumed to exist.

$$\frac{\pi_i p_{ij}\overline{s_{ij}^o}}{\sum_{i=1}^{J}\sum_{j=1}^{J}\pi_i p_{ij}\overline{s_{ij}^o}}, \qquad (i,j)\in\Pi^s. \tag{7.34}$$

Furthermore the long-run fraction of time that the system is in a switchover period is $1-\rho$. Hence we obtain

$$\tilde{q}^{(i,j)} = (1-\rho)\frac{\pi_i p_{ij}\overline{s_{ij}^o}}{\sum_{i=1}^{J}\sum_{j=1}^{J}\pi_i p_{ij}\overline{s_{ij}^o}}, \qquad (i,j)\in\Pi^s. \tag{7.35}$$

The expected remaining service time of a customer being served given that the current period is $k\in\Pi$ is equal to $\overline{s_k^2}/(2\mathrm{E}[S_k])$, and the expected value of the remaining switchover period given that the current period is $(i,j)\in\Pi^s$ is equal to $\overline{s_{ij}^{o2}}/(2\overline{s_{ij}^o})$. Then we have

$$\tilde{r}^k = \left(\frac{\overline{s_k^2}}{2\mathrm{E}[S_k]}\right)\tilde{q}^k = \frac{\lambda_k\overline{s_k^2}}{2}, \quad k\in\Pi, \tag{7.36}$$

$$\tilde{r}^{(i,j)} = \left(\frac{\overline{s_{ij}^{o2}}}{2\overline{s_{ij}^o}}\right)\tilde{q}^{(i,j)} = (1-\rho)\frac{\pi_i p_{ij}\overline{s_{ij}^{o2}}}{2\sum_{i=1}^{J}\sum_{j=1}^{J}\pi_i p_{ij}\overline{s_{ij}^o}}, \quad (i,j)\in\Pi^s. \tag{7.37}$$

From the results in the previous sections and the PASTA property, we have

$$\begin{aligned}\bar{H}_j(\kappa_0,k) &\\ &= \begin{cases}\tilde{r}^{\kappa_0}\varphi(\kappa_0,j,k)+(\tilde{\mathbf{g}}^{\kappa_0},\tilde{\mathbf{n}}^{\kappa_0})\mathbf{h}_{00}(\kappa_0,j,k)+\tilde{q}^{\kappa_0}h_{01}(\kappa_0,j,k), & k\in\Pi\\ \tilde{r}^{\kappa_0}\varphi(\kappa_0,j,k)+\tilde{q}^{\kappa_0}h_{01}(\kappa_0,j,k), & k\in\Pi^s,\end{cases}\end{aligned} \tag{7.38}$$

$$\bar{F}_j(\kappa_0) = \tilde{r}^{\kappa_0}\psi(\kappa_0,j)+(\tilde{\mathbf{g}}^{\kappa_0},\tilde{\mathbf{n}}^{\kappa_0})\mathbf{f}(\kappa_0,j) \tag{7.39}$$

for $j\in\Pi$ and $\kappa_0\in\Pi\cup\Pi^s$. Then from the generalized version of Little's formula ($H=\lambda G$) [27], we have

$$\begin{aligned}\tilde{n}_j^k &= \lambda_j\sum_{\kappa_0\in\Pi\cup\Pi^s}\bar{H}_j(\kappa_0,k),\\ \tilde{g}_j &= \lambda_j\sum_{\kappa_0\in\Pi\cup\Pi^s}\bar{F}_j(\kappa_0),\end{aligned} \qquad j\in\Pi \quad \text{and} \quad k\in\Pi\cup\Pi^s, \tag{7.40}$$

where $\tilde{g}_j = \sum_{k\in\Pi\cup\Pi^s}\tilde{g}_j^k$. Furthermore it can be shown that

$$\tilde{g}_j^k = \begin{cases}\tilde{g}_j, & k=j,\\ 0, & k\neq j,\end{cases} \qquad j\in\Pi \quad \text{and} \quad k\in\Pi\cup\Pi^s. \tag{7.41}$$

Then we obtain the following set of linear equations for the average numbers of customers in the system.

$$\tilde{n}_j^k = \begin{cases} \lambda_j \sum_{\kappa_0 \in \Pi \cup \Pi^s} \{\tilde{r}^{\kappa_0}\varphi(\kappa_0, j, k) + \tilde{q}^{\kappa_0} h_{01}(\kappa_0, j, k) \\ \qquad\qquad + (\tilde{\mathbf{g}}^{\kappa_0}, \tilde{\mathbf{n}}^{\kappa_0})\mathbf{h}_{00}(\kappa_0, j, k)\}, & k \in \Pi \\ \lambda_j \sum_{\kappa_0 \in \Pi \cup \Pi^s} \{\tilde{r}^{\kappa_0}\varphi(\kappa_0, j, k) + \tilde{q}^{\kappa_0} h_{01}(\kappa_0, j, k)\}, & k \in \Pi^s, \end{cases} \tag{7.42}$$

$$\tilde{g}_j^k = \begin{cases} \lambda_j \sum_{\kappa_0 \in \Pi \cup \Pi^s} \{\tilde{r}^{\kappa_0}\psi(\kappa_0, j) + (\tilde{\mathbf{g}}^{\kappa_0}, \tilde{\mathbf{n}}^{\kappa_0})\mathbf{f}(\kappa_0, j)\}, & k = j \\ 0, & k \neq j \text{ or } k \in \Pi^s \end{cases} \tag{7.43}$$

for $j \in \Pi$ and $k \in \Pi \cup \Pi^s$. Then we finally obtain the following proposition.

Proposition 7.4. *The mean waiting time of j-customers ($j = 1, \ldots, J$) is given by*

$$\bar{w}_j = \sum_{\kappa_0 \in \Pi \cup \Pi^s} \left\{ \sum_{k \in \Pi \cup \Pi^s} \bar{H}_j(\kappa_0, k) + \bar{F}_j(\kappa_0) \right\} = \frac{1}{\lambda_j} \left(\tilde{g}_j^j + \sum_{k \in \Pi \cup \Pi^s} \tilde{n}_j^k \right), \tag{7.44}$$

where $\tilde{g}_j^j$ and $\tilde{n}_j^k$ ($j \in \Pi$; $k \in \Pi \cup \Pi^s$) can be obtained by solving the set of (7.42) and (7.43).

7.6 Computational Complexity

We now evaluate the computational complexity to calculate the mean waiting times. In Sect. 7.4 calculation of the coefficients $\mathbf{h}_{10}(\kappa_0, j, k)$ $(\kappa_0, j, k \in \Pi)$ takes much time. Then from (7.26) we have

$$\begin{pmatrix} \mathbf{h}_{10}(1, j, k) \\ \mathbf{h}_{10}(2, j, k) \\ \vdots \\ \mathbf{h}_{10}(J, j, k) \end{pmatrix} = (\mathbf{I} - \mathbf{I}(j)\mathbf{U}\mathbf{Q})^{-1}\mathbf{I}(j) \begin{pmatrix} \mathbf{h}_{10}^0(1, j, k) \\ \mathbf{h}_{10}^0(2, j, k) \\ \vdots \\ \mathbf{h}_{10}^0(J, j, k) \end{pmatrix},$$

where $\mathbf{I} \in \mathscr{R}^{2J^2 \times 2J^2}$ and $\mathbf{I}_0 \in \mathscr{R}^{2J \times 2J}$ are identity matrices, and where

$$\mathbf{I}(j) = \mathrm{diag}(\mathbf{I}_0, \ldots, \mathbf{I}_0, \underbrace{\mathbf{O}}_{j^{th}\text{ place}}, \mathbf{I}_0, \ldots, \mathbf{I}_0) \in \mathscr{R}^{2J^2 \times 2J^2},$$

$$\mathbf{Q} = (p_{i,j}\mathbf{I}_0 : i, j = 1, \ldots, J) \in \mathscr{R}^{2J^2 \times 2J^2},$$

$$\mathbf{U} = \mathrm{diag}(\mathbf{U}_1(j) : j = 1, \ldots, J) \in \mathscr{R}^{2J^2 \times 2J^2}.$$

The calculation of the J inverse matrices $(\mathbf{I}-\mathbf{I}(j)\mathbf{UQ})^{-1}\mathbf{I}(j) \in \mathscr{R}^{2J^2\times 2J^2}$, $(j \in \Pi)$ takes $J \times O(J^6)$ numerical operations,[7] and the whole calculation of the products of the inverse matrices and the right end vectors of vectors $\{\mathbf{h}_{10}^0(\kappa_0, j, k) : \kappa_0 \in \Pi\}$, $(j,k \in \Pi)$ take $O(J^6)$ numerical operations. The calculations of the other constants in Sect. 7.4 take at most $O(J^5)$ numerical operations. In Sect. 7.5 it takes much time to solve (7.42) and (7.43). Because the set of these equations essentially has $J(J+1)$ unknowns, $O(J^6)$ numerical operations are required in order to solve them. The other calculations in this section take at most $O(J^5)$ numerical operations. Hence the overall complexity of our method is primarily $O(J^7)$ numerical operations.

The primal algorithm has somewhat excessive computational complexity, therefore we would like to reduce it. As noted above, much of the computational complexity comes from the calculations of the constants $\mathbf{h}_{10}(\kappa_0, j, k)$ and the calculations of the steady-state values from (7.42) and (7.43).

7.6.1 Reduction of Calculations of $\mathbf{h}_{10}(\cdot)$

This reduction has three steps.

First Reduction Step:

We can reduce the computational complexity by checking the following facts. Because it can be shown from (7.15) and (7.16) that $H_j^0(\mathbf{Y}, e, l, k)$ for $\kappa_0, k \in \Pi$ and $l > 0$ is not affected by the vector $\mathbf{g}$ of the numbers of customers in the service facility, the elements in the upper half of $\mathbf{h}_{10}^0(\kappa_0, j, k)$ in (7.23) are 0. Similarly, because it can be shown from (7.19) and (7.20) that the conditional expectation of $\mathbf{n}(\tau_{l+1}^e)$ for $\kappa_0 \in \Pi$ and $l > 0$ is not affected by $\mathbf{g}$, and because $\mathbf{g}(\tau_{l+1}^e) = \mathbf{0}$, the elements in the upper half and the left half of $\mathbf{U}_1(\kappa_0)$ in (7.25) are 0. That is, we have

$$\mathbf{h}_{10}^0(\kappa_0, j, k) = \begin{pmatrix} \mathbf{0} \\ \mathbf{h}_{10}^{0*}(\kappa_0, j, k) \end{pmatrix}, \qquad \mathbf{U}_1(\kappa_0) = \begin{pmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{U}_1^*(\kappa_0) \end{pmatrix},$$

where $\mathbf{h}_{10}^{0*}(\kappa_0, j, k) \in \mathscr{R}^{J\times 1}$, $\mathbf{U}_1^*(\kappa_0) \in \mathscr{R}^{J\times J}$, and then the size of (7.26) can be reduced by half. Let $\mathbf{h}_{10}^*(\kappa_0, j, k) \in \mathscr{R}^{J\times 1}$ be the vector composed of the lower half elements of $\mathbf{h}_{10}(\kappa_0, j, k)$, and we have the following reduced version of (7.26).

$$\mathbf{h}_{10}^*(\kappa_0, j, k) = \begin{cases} \mathbf{h}_{10}^{0*}(\kappa_0, j, k) + \mathbf{U}_1^*(\kappa_0) \displaystyle\sum_{\kappa_1 \in \Pi\setminus\{j\}} p_{\kappa_0\kappa_1}\mathbf{h}_{10}^*(\kappa_1, j, k), \\ \qquad\qquad\qquad\qquad \kappa_0 \neq j,\ \kappa_0 \in \Pi,\ k \in \Pi \\ \mathbf{0}, \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{7.45}$$

for $\kappa_0, k \in \Pi \cup \Pi^s$ and $j \in \Pi$.

[7] For simplicity, we estimate that an $n \times n$ matrix can be inverted in $O(n^3)$ numerical operations.

Second Reduction Step:

We reduce the calculations by using sparsity of the constants $\mathbf{h}_{10}^{0*}(\kappa_0, j, k)$ and $\mathbf{U}_1^*(\kappa_0)$ in (7.45). From (7.15), (7.16), and (7.23), for $\kappa_0 \in \Pi,\ l > 0,\ k \in \Pi$, we have

$$\mathbf{h}_{10}^{0*}(\kappa_0, j, k) = \begin{cases} \mathbf{e}'_{\kappa_0}\delta(\kappa_0), & \kappa_0 = k,\ \kappa_0 \neq j \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (j \in \Pi), \tag{7.46}$$

where $\mathbf{e}'_{\kappa_0}$ is the transpose of $\mathbf{e}_{\kappa_0} = (0, \ldots, 0, \underbrace{1}_{\kappa_0^{th}\text{ place}}, 0, \ldots, 0)$ defined in Sect. 7.3, and

where

$$\delta(\kappa_0) = \begin{cases} \mathrm{E}[S_{\kappa_0}], & \kappa_0 \in \mathscr{H}_g \\ \mathrm{E}[S_{\kappa_0}]/(1-\rho_{\kappa_0}), & \kappa_0 \in \mathscr{H}_e. \end{cases}$$

From (7.19), (7.20), and (7.25), it can be shown that

$$\mathbf{U}_1^*(\kappa_0) = \begin{pmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ u_1^*(\kappa_0) & \cdots & u_{\kappa_0-1}^*(\kappa_0) & u_{\kappa_0}^*(\kappa_0) & u_{\kappa_0+1}^*(\kappa_0) & \cdots & u_J^*(\kappa_0) \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

where $u_m^*(\kappa_0) = \lambda_m \delta(\kappa_0)\mathbf{1}\{m \neq \kappa_0 \text{ or } \kappa_0 \in \mathscr{H}_g\}$, $(m, \kappa_0 \in \Pi)$.

Let

$$\xi_{10}^*(\kappa_0, j, k) = \begin{cases} \sum\limits_{\kappa_1 \in \Pi \setminus \{j\}} p_{\kappa_0\kappa_1}\mathbf{h}_{10}^*(\kappa_1, j, k), & \kappa_0 \neq j \\ \mathbf{0}, & \kappa_0 = j \end{cases} \quad (\kappa_0, j, k \in \Pi).$$

Then from (7.45) and (7.46), we have

$$\mathbf{h}_{10}^*(\kappa_0, j, k) = \mathbf{e}'_{\kappa_0}\delta(\kappa_0)\mathbf{1}\{\kappa_0 = k, \kappa_0 \neq j\} + \mathbf{U}_1^*(\kappa_0)\xi_{10}^*(\kappa_0, j, k)$$

for $\kappa_0, j, k \in \Pi$. Now we define the following notation.

- For any vector $\mathbf{a}$, let $\mathbf{a}|_m$ be its mth element.

Then we have

$$\begin{aligned} &\mathbf{h}_{10}^*(\kappa_0, j, k)|_m \\ &= \begin{cases} \xi_{10}^*(\kappa_0, j, k)|_m, & m \neq \kappa_0 \\ \delta(\kappa_0)\mathbf{1}\{\kappa_0 = k,\ \kappa_0 \neq j\} + \sum\limits_{l=1}^{J} u_l^*(\kappa_0)\xi_{10}^*(\kappa_0, j, k)|_l, & m = \kappa_0, \end{cases} \end{aligned} \tag{7.47}$$

$$\xi^*_{10}(\kappa_0, j,k)|_m \tag{7.48}$$
$$= \begin{cases} \sum\limits_{\kappa_1 \in \Pi\setminus\{j,m\}} p_{\kappa_0\kappa_1}\xi^*_{10}(\kappa_1, j,k)|_m + p_{\kappa_0 m}\mathbf{h}^*_{10}(m, j,k)|_m, & m \neq j,\ \kappa_0 \neq j \\ \sum\limits_{\kappa_1 \in \Pi\setminus\{j\}} p_{\kappa_0\kappa_1}\xi^*_{10}(\kappa_1, j,k)|_j, & m = j,\ \kappa_0 \neq j \\ 0, & \kappa_0 = j \end{cases}$$

for $m = 1,\dots,J,\ \kappa_0, j,k \in \Pi$. Let

$$\xi^*_{10}(j,k)_m = \begin{pmatrix} \xi^*_{10}(1, j,k)|_m \\ \vdots \\ \xi^*_{10}(J, j,k)|_m \end{pmatrix} \in \mathscr{R}^{J\times 1}, \qquad \mathbf{p}(j)_m = \begin{pmatrix} p_{1,m} \\ \vdots \\ p_{j-1,m} \\ 0 \\ p_{j+1,m} \\ \vdots \\ p_{J,m} \end{pmatrix} \in \mathscr{R}^{J\times 1},$$

$$\mathbf{I}_0(m) = \mathrm{diag}(1,\cdots,1,\underbrace{0}_{m^{th}\text{ place}},1,\cdots,1) \in \mathscr{R}^{J\times J},$$

$$\mathbf{P}(j) = \big(\mathbf{p}(j)_1,\cdots,\mathbf{p}(j)_{j-1},\mathbf{0},\mathbf{p}(j)_{j+1},\cdots,\mathbf{p}(j)_J\big) \in \mathscr{R}^{J\times J}$$

for $m = 1,\dots,J,\ j,k \in \Pi$. Then from (7.48), we have

$$\xi^*_{10}(j,k)_m = \begin{cases} \mathbf{P}(j)\mathbf{I}_0(m)\xi^*_{10}(j,k)_m + \mathbf{p}(j)_m\mathbf{h}^*_{10}(m, j,k)|_m, & m \neq j \\ \mathbf{P}(j)\xi^*_{10}(j,k)_j, & m = j \end{cases}$$

or

$$\xi^*_{10}(j,k)_m = \begin{cases} (\mathbf{I} - \mathbf{P}(j)\mathbf{I}_0(m))^{-1}\mathbf{p}(j)_m\mathbf{h}^*_{10}(m, j,k)|_m, & m \neq j \\ \mathbf{0}, & m = j. \end{cases} \tag{7.49}$$

Now let

$$\eta^*_{10}(j,k) = \begin{pmatrix} \mathbf{h}^*_{10}(1, j,k)|_1 \\ \vdots \\ \mathbf{h}^*_{10}(J, j,k)|_J \end{pmatrix} \in \mathscr{R}^{J\times 1},$$

$$\mathbf{U}^*(j)_m = \mathrm{diag}(u^*_m(1),\cdots,u^*_m(j-1),0,u^*_m(j+1),\cdots,u^*_m(J)) \in \mathscr{R}^{J\times J},$$

$$\delta(k) = (0,\cdots,0,\underbrace{\delta(k)}_{k^{th}\text{ place}},0,\cdots,0)' \in \mathscr{R}^{J\times 1}.$$

Then from (7.47) and (7.49), we have

$$\begin{aligned}\eta^*_{10}(j,k) &= \delta(k)\mathbf{1}\{k \neq j\} + \sum_{m=1}^{J} \mathbf{U}^*(j)_m \xi^*_{10}(j,k)_m \\ &= \delta(k)\mathbf{1}\{k \neq j\} + \sum_{m\neq j} \mathbf{U}^*(j)_m \mathbf{q}(j)_m \mathbf{h}^*_{10}(m,j,k)|_m,\end{aligned}$$

where $\mathbf{q}(j)_m = (\mathbf{I} - \mathbf{P}(j)\mathbf{I}_0(m))^{-1}\mathbf{p}(j)_m$. Let

$$\begin{aligned}\mathscr{U}^*(j) = \big(&\mathbf{U}^*(j)_1\mathbf{q}(j)_1, \ldots, \mathbf{U}^*(j)_{j-1}\mathbf{q}(j)_{j-1}, \mathbf{0}, \\ &\mathbf{U}^*(j)_{j+1}\mathbf{q}(j)_{j+1}, \ldots, \mathbf{U}^*(j)_J\mathbf{q}(j)_J\big) \in \mathscr{R}^{J\times J}.\end{aligned} \tag{7.50}$$

Then we have

$$\eta^*_{10}(j,k) = \delta(k)\mathbf{1}\{k \neq j\} + \mathscr{U}^*(j)\eta^*_{10}(j,k), \qquad (j,k \in \Pi). \tag{7.51}$$

Algorithm for the second reduction: Repeat the following steps for $j = 1, \ldots, J$.

1. Solve $(\mathbf{I} - \mathbf{P}(j)\mathbf{I}_0(m))\mathbf{q}(j)_m = \mathbf{p}(j)_m$ to obtain $\mathbf{q}(j)_m$ for $m \neq j$.
2. Set matrix $\mathscr{U}^*(j)$ defined in (7.50).
3. Solve the set of the equations given by (7.51) to obtain $\eta^*_{10}(j,k)$ for $k \neq j$.
4. From (7.49), $\xi^*_{10}(j,k)_m = \mathbf{q}(j)_m\mathbf{h}^*_{10}(m,j,k)|_m = \mathbf{q}(j)_m\eta^*_{10}(j,k)|_m$ for $m \neq j$[8].
5. From the definition of $\eta^*_{10}(\cdot)$ and (7.47), we have[9]

$$\mathbf{h}^*_{10}(\kappa_0,j,k)|_m = \begin{cases} \eta^*_{10}(j,k)|_{\kappa_0}, & m = \kappa_0 \\ \xi^*_{10}(\kappa_0,j,k)|_m = \mathbf{q}(j)_m|_{\kappa_0}\mathbf{h}^*_{10}(m,j,k)|_m, & m \neq \kappa_0. \end{cases} \tag{7.52}$$

Third Reduction Step:

The computational effort in the algorithm for the second reduction can be further reduced in the following manner. It can be easily shown that $(\mathbf{I} - \mathbf{P}(j)\mathbf{I}_0(m))\mathbf{q}(j)_m = \mathbf{p}(j)_m$ for $m \neq j$ can be written as follows:

$$(\mathbf{I} - \mathbf{P}(j))\mathbf{q}(j)_m = \mathbf{p}(j)_m(1 - \mathbf{q}(j)_m|_m),$$

where $\mathbf{q}(j)_m|_m$ is the mth element of the vector $\mathbf{q}(j)_m$. Hence we have

$$\mathbf{q}(j)_m = (\mathbf{I} - \mathbf{P}(j))^{-1}\mathbf{p}(j)_m(1 - \mathbf{q}(j)_m|_m).$$

Then it can be easily shown that

$$\mathbf{q}(j)_m|_m = \mathbf{q}'(j)_m|_m(1 + \mathbf{q}'(j)_m|_m)^{-1},$$

[8] Because $\mathbf{h}^*_{10}(j,j,k)|_j = \eta^*_{10}(j,k)|_j = 0$, $\xi^*_{10}(j,k)_m = \mathbf{q}(j)_m\mathbf{h}^*_{10}(m,j,k)|_m$ is also true for $m = j$.

[9] $\eta^*_{10}(j,k)|_{\kappa_0}$ and $\mathbf{q}(j)_m|_{\kappa_0}$ are the κ_0th elements of $\eta^*_{10}(j,k)$ and $\mathbf{q}(j)_m$, respectively.

where $\mathbf{q}'(j)_m|_m$ is the mth element of the vector $\mathbf{q}'(j)_m = (\mathbf{I} - \mathbf{P}(j))^{-1}\mathbf{p}(j)_m$. The first step 1 of the algorithm for the second reduction then can be arranged as

1′. Solve $(\mathbf{I} - \mathbf{P}(j))\mathbf{q}'(j)_m = \mathbf{p}(j)_m$ to obtain $\mathbf{q}'(j)_m$ for $m \neq j$. Then set

$$\mathbf{q}(j)_m|_m = \mathbf{q}'(j)_m|_m(1 + \mathbf{q}'(j)_m|_m)^{-1}, \qquad \mathbf{q}(j)_m = \mathbf{q}'(j)_m(1 - \mathbf{q}(j)_m|_m).$$

The computational complexity of the algorithm is evaluated later.

7.6.2 Reduction of Calculations of Steady-State Values

Because we cannot further reduce the number of the steady-state equations, we would like to solve them by a successive approximation instead of directly solving them. Because it takes much computational effort to apply the original equations (7.42) and (7.43) to the approximation, we would like to reduce it by arranging coefficients $\mathbf{h}_{00}(\cdot)$ as follows.

From (7.15) (7.16), and (7.23), we have

$$\mathbf{h}^0_{00}(\kappa_0, j, k) = \begin{pmatrix} * \\ \mathbf{0} \end{pmatrix}, \qquad (\kappa_0, j, k \in \Pi).$$

That is, $H^0_j(\mathbf{Y}, e, l, k)$ for $\kappa_0, k \in \Pi$ and $l = 0$ is not affected by the number of customers in the waiting room $\mathbf{n}$. From (7.19), (7.20), (7.22), and (7.25), we have

$$\mathbf{U}_0(\kappa_0) = \begin{pmatrix} \mathbf{O} & * \\ \mathbf{O} & \mathbf{U}^*_{01}(\kappa_0) \end{pmatrix}, \qquad (\kappa_0 \in \Pi),$$

where

$$\mathbf{U}^*_{01}(\kappa_0) = \operatorname{diag}(1, \ldots, 1, \underbrace{\mathbf{1}\{\kappa_0 \in \mathscr{H}_g\}}_{\kappa_0^{th}\text{ place}}, 1, \ldots, 1) \in \mathscr{R}^{J \times J}.$$

Let $\mathbf{h}^*_{00}(\kappa_0, j, k) \in \mathscr{R}^{J \times 1}$ be the lower half of $\mathbf{h}_{00}(\kappa_0, j, k)$ for $\kappa_0, j, k \in \Pi$. Then from the definition of $\mathbf{h}_{00}(\kappa_0, j, k)$ in Sect. 7.4, we have

$$\mathbf{h}^*_{00}(\kappa_0, j, k) = \mathbf{U}^*_{01}(\kappa_0) \sum_{\kappa_1 \in \Pi \setminus \{j\}} p_{\kappa_0 \kappa_1} \mathbf{h}^*_{10}(\kappa_1, j, k), \qquad (\kappa_0 \neq j \text{ or } j \in \mathscr{H}_g)$$

and its mth element is given by

$$\mathbf{h}^*_{00}(\kappa_0, j, k)|_m = \begin{cases} \displaystyle\sum_{\kappa_1 \in \Pi \setminus \{j\}} p_{\kappa_0 \kappa_1} \mathbf{h}^*_{10}(\kappa_1, j, k)|_m, & m \neq \kappa_0 \text{ or } \kappa_0 \in \mathscr{H}_g \\ 0, & m = \kappa_0 \in \mathscr{H}_e \end{cases}$$

for $\kappa_0 \neq j$ or $j \in \mathscr{H}_g$ $(\kappa_0, j, k \in \Pi)$. ($\mathbf{h}^*_{00}(\kappa_0, j, k)|_m = 0$ for all m when $\kappa_0 = j \in \mathscr{H}_e$.)

Furthermore from (7.52), it can be shown that

$$\sum_{\kappa_1 \in \Pi \setminus \{j\}} p_{\kappa_0 \kappa_1} \mathbf{h}^*_{10}(\kappa_1, j, k)|_m = q^0(\kappa_0, j)_m \mathbf{h}^*_{10}(m, j, k)|_m \tag{7.53}$$

for $\kappa_0, j, k \in \Pi$, where

$$q^0(\kappa_0, j)_m = \begin{cases} \sum_{\kappa_1 \in \Pi \setminus \{j,m\}} p_{\kappa_0 \kappa_1} \mathbf{q}(j)_m|_{\kappa_1} + p_{\kappa_0 m}, & m \neq j \\ \sum_{\kappa_1 \in \Pi \setminus \{j\}} p_{\kappa_0 \kappa_1} \mathbf{q}(j)_j|_{\kappa_1}, & m = j. \end{cases} \tag{7.54}$$

Then we have the final expression for $\mathbf{h}^*_{00}$:

$$\mathbf{h}^*_{00}(\kappa_0, j, k)|_m = q^*(\kappa_0, j)_m \mathbf{h}^*_{10}(m, j, k)|_m \tag{7.55}$$

for $\kappa_0, j, k, m \in \Pi$, where

$$q^*(\kappa_0, j)_m = \begin{cases} q^0(\kappa_0, j)_m, & (m \neq \kappa_0 \text{ or } \kappa_0 \in \mathscr{H}_g) \text{ } and \text{ } (\kappa_0 \neq j \text{ or } j \in \mathscr{H}_g) \\ 0, & \text{otherwise.} \end{cases} \tag{7.56}$$

Then from (7.42) and (7.43), it can be easily shown that the steady-state numbers of customers satisfy the following equations:

$$\tilde{n}^k_j = \begin{cases} \tilde{\varphi}^k_{hj} + \lambda_j \sum_{\kappa_0 \in \Pi} \tilde{g}^{\kappa_0}_{\kappa_0} \mathbf{h}_{00}(\kappa_0, j, k)|_{\kappa_0} \\ \qquad + \lambda_j \sum_{m=1}^{J} \tilde{n}_{qm}(j) \mathbf{h}^*_{10}(m, j, k)|_m, & k \in \Pi \\ \tilde{\varphi}^k_j, & k \in \Pi^s, \end{cases} \tag{7.57}$$

$$\tilde{g}^j_j = \tilde{\psi}_{fj} + \lambda_j \sum_{\kappa_0 \in \Pi} \tilde{g}^{\kappa_0}_{\kappa_0} \mathbf{f}(\kappa_0, j)|_{\kappa_0} + \lambda_j \sum_{\kappa_0 \in \Pi} \sum_{m=1}^{J} \tilde{n}^{\kappa_0}_m \mathbf{f}(\kappa_0, j)|_{J+m}, \tag{7.58}$$

$$\tilde{n}_{qm}(j) = \sum_{\kappa_0 \in \Pi} \tilde{n}^{\kappa_0}_m q^*(\kappa_0, j)_m \tag{7.59}$$

for $k \in \Pi \cup \Pi^s$ and $j, m \in \Pi$, where

$$\begin{aligned} \tilde{\varphi}^k_j &= \lambda_j \sum_{\kappa_0 \in \Pi \cup \Pi^s} \{\tilde{r}^{\kappa_0} \varphi(\kappa_0, j, k) + \tilde{q}^{\kappa_0} h_{01}(\kappa_0, j, k)\}, \\ \tilde{\varphi}^k &= \left(\tilde{\varphi}^k_j : j = 1, \ldots, J\right) \in \mathscr{R}^{1 \times J}, \\ \tilde{\varphi}^k_{hj} &= \tilde{\varphi}^k_j + \lambda_j \sum_{\kappa_0 \in \Pi^s} (\mathbf{0}, \tilde{\varphi}^{\kappa_0}) \mathbf{h}_{00}(\kappa_0, j, k), \qquad (k \in \Pi \text{ for this case}), \\ \tilde{\psi}_{fj} &= \lambda_j \sum_{\kappa_0 \in \Pi \cup \Pi^s} \{\tilde{r}^{\kappa_0} \psi(\kappa_0, j)\} + \lambda_j \sum_{\kappa_0 \in \Pi^s} (\mathbf{0}, \tilde{\varphi}^{\kappa_0}) \mathbf{f}(\kappa_0, j). \end{aligned} \tag{7.60}$$

From the equations, we can construct a successive approximation algorithm for the steady-state values. Note that $\tilde{n}^k_j$ $(k \in \Pi^s,\ j \in \Pi)$ can be directly calculated in advance from the known constants.

Algorithm for calculating the steady-state values by successive approximation

1. Set $s = 0$ and the initial values of $\tilde{n}_j^{k(0)}, \tilde{g}_j^{j(0)}, \tilde{n}_{qm}^{(0)}(j)$ for $j,k,m \in \Pi$.
2. Calculate $\tilde{n}_j^{k(s+1)}, \tilde{g}_j^{j(s+1)}, \tilde{n}_{qm}^{(s+1)}(j)$ for $j,k,m \in \Pi$ from the set of equations:

$$\tilde{n}_j^{k(s+1)} = \tilde{\varphi}_{hj}^k + \lambda_j \sum_{\kappa_0 \in \Pi} \tilde{g}_{\kappa_0}^{\kappa_0(s)} \mathbf{h}_{00}(\kappa_0, j, k)|_{\kappa_0} + \lambda_j \sum_{m=1}^{J} \tilde{n}_{qm}^{(s)}(j) \mathbf{h}_{10}^{*}(m, j, k)|_m,$$

$$\tilde{g}_j^{j(s+1)} = \tilde{\psi}_{fj} + \lambda_j \sum_{\kappa_0 \in \Pi} \tilde{g}_{\kappa_0}^{\kappa_0(s)} \mathbf{f}(\kappa_0, j)|_{\kappa_0} + \lambda_j \sum_{\kappa_0 \in \Pi} \sum_{m=1}^{J} \tilde{n}_m^{\kappa_0(s)} \mathbf{f}(\kappa_0, j)|_{J+m},$$

$$\tilde{n}_{qm}^{(s+1)}(j) = \sum_{\kappa_0 \in \Pi} \tilde{n}_m^{\kappa_0(s+1)} q^{*}(\kappa_0, j)_m.$$

3. If these values are considered to converge, then stop. Otherwise, let $s \leftarrow s+1$ and go to step 2.

Note. We can show (1) the uniqueness of the solution of (7.42) and (7.43), and (2) the convergence of the values obtained by the successive approximation method to the unique solution (under the assumption that these steady-state average values exist).

7.6.3 Evaluation of Computational Complexity

We now evaluate the computational complexity after the reductions. After applying the third reduction step, in order to derive $\mathbf{h}_{10}(\cdot)$, we are essentially required to solve the J sets of the $O(J)$ linear equations related to the equations $(\mathbf{I} - \mathbf{P}(j))\mathbf{q}'(j)_m = \mathbf{p}(j)_m$, and required to solve the J sets of the $O(J)$ linear equations related to (7.51). And a careful estimation shows that the other calculations require at most $O(J^4)$ numerical operations. Then it can be easily shown that only $O(J^4)$ numerical operations are required in order to calculate the constants $\mathbf{h}_{10}^{*}(\kappa_0, j, k)$ and $\mathbf{q}(j)_m$ $(\kappa_0, j, k, m \in \Pi)$. Hence if we directly solve the steady-state equations (7.42) and (7.43) by inverting the coefficient matrix after applying the third reduction step, $O(J^6)$ numerical operations are required in order to calculate the mean waiting times for all stations.

Then for the successive approximation of the steady-state values $(\tilde{\mathbf{g}}^k, \tilde{\mathbf{n}}^k)$, it is clear that $O(J^3)$ numerical operations are required in order to calculate the values at each iterative step. And it can be shown that calculations of the other coefficients $(\{\tilde{\varphi}_{hj}^k : j,k \in \Pi\}, \{\tilde{\varphi}_j^k : k \in \Pi \cup \Pi^s,\ j \in \Pi\}, \{\mathbf{h}_{00}(\kappa_0, j, k)|_{\kappa_0} : \kappa_0, j, k \in \Pi\}, \{\tilde{\psi}_{fj} : j \in \Pi\}, \{\mathbf{f}(\kappa_0, j) : \kappa_0, j \in \Pi\}, \{q^{*}(\kappa_0, j)_m : \kappa_0, j, m \in \Pi\})$ which appear in (7.57)–(7.60) require $O(J^4)$ numerical operations.

Hence if we obtain the mean waiting times for all stations after applying the third reduction step and the successive approximation for the steady-state values, $O(J^4) + O(J^3 N)$ numerical operations are required where $N = N_{J,\rho,\varepsilon}$ is the number of iterations of the approximation that depends on the number of stations J, the resource utilization ρ, and the required accuracy ε[10].

7.6.4 Comparison of Computational Times by Examples

Now we compare our functional computation method with the buffer occupancy method by actually measuring their running times to compute the average waiting times in the systems with $J = 40$ stations and $J = 80$ stations. Half of the stations take the gated rule and the other stations take the exhaustive rule. In order to make graphs for the running times in each system by changing the resource utilization ρ, the arrival rates are varied. The service times, the switchover times, and the switching probabilities are fixed. The algorithms that adopt the following methods are compared.

Ours 1: Our functional computation method that calculates the steady-state values by directly solving the equations (i.e., inverting their coefficient matrix)

Ours 2: Our functional computation method that calculates the steady-state values by the successive approximation

B.O.: The buffer occupancy method that calculates second moments of the buffer occupancy variables by a successive approximation.

In Figs. 7.1 and 7.2, "Ours 2-1" and "Ours 2-2" denote our second method "Ours 2" with $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-8}$, respectively,[11] and "B.O.1" and "B.O.2" denote the buffer occupancy method "B.O." with $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-8}$, respectively. Although the running times of "Ours 1" do not depend on the resource utilization, they are somewhat greater than those of "Ours 2." "Ours 2" takes almost constant running times until the resource utilization reaches about 0.9. It results from the fact that when ρ is less than the value, the number of iterations N is relatively small and the computational complexity of "Ours 2" is approximately $O(J^4)$. When ρ approaches 1, N grows rapidly and its running times also grow rapidly. The numbers of iterations for the approximation methods are given in Tables 7.1 and 7.2. We see from the tables that the numbers of iterations of "B.O." are fairly (10 or more times) greater than those of "Ours 2." This may be caused by the difference between the numbers of variables in the steady-state equations; that is, "Ours 2" has only $O(J^2)$

---

[10] As noted in Sect. 7.1, the computational complexity of the buffer occupancy method that uses an approximation is $O(J^4 N')$ where N' is the number of its iterations.

[11] When $\left|\sum_{j=1}^{J} \rho_j \bar{w}_j^{(s)} - \sum_{j=1}^{J} \rho_j \bar{w}_j^{(s-1)}\right| < \varepsilon$, the successive approximation methods stop, where $\{\bar{w}_j^{(s)}\}$ is a set of the mean waiting times obtained at their sth iterative step and ε is a required accuracy. The used CPU is the AMD Athlon 64 X 2 4400+ with 4 GB memories, and the programming language is Intel Visual FORTRAN with the IMSL Library.

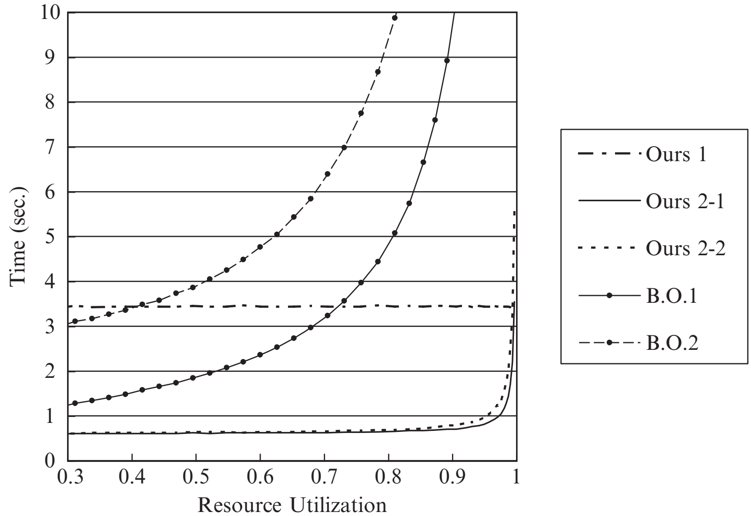

Fig. 7.1 Running times for computing the mean waiting times in the system with $J = 40$.

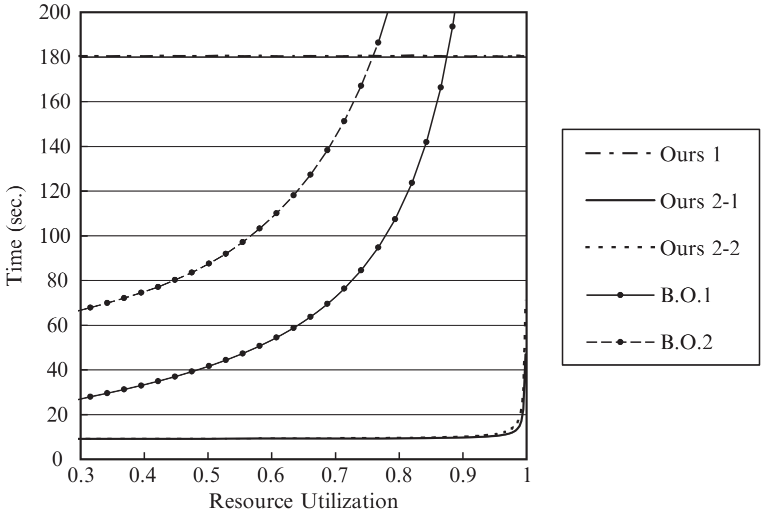

Fig. 7.2 Running times for computing the mean waiting times in the system with $J = 80$.

variables in contrast to "B.O." which has $O(J^3)$ variables. Furthermore for the buffer occupancy method, because $O(J^4)$ operations per iteration are required, its running times are greater than those of "Ours 2." These differences become large as the system is congested (i.e., when ρ is large).

Table 7.1 Numbers of iterations (N and N') for the system with $J = 40$.

	Ours 2 (N)				B.O. (N')			
	Required Accuracies (ε)				Required Accuracies (ε)			
ρ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
0.3114	5	8	11	14	86	230	399	587
0.5213	9	13	18	23	166	363	566	772
0.7574	19	29	39	49	383	755	1126	1497
0.9057	50	76	102	128	1051	1992	2934	3875
0.9568	110	167	224	281	2339	4389	6440	8490
0.9899	471	714	957	1200	10102	18844	27585	36326

Table 7.2 Numbers of iterations (N and N') for the system with $J = 80$.

	Ours 2 (N)				B.O. (N')			
	Required Accuracies (ε)				Required Accuracies (ε)			
ρ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
0.3154	6	9	11	14	177	463	786	1152
0.5014	9	13	18	22	318	701	1092	1490
0.7405	19	28	37	46	729	1440	2152	2863
0.8986	49	73	96	119	2006	3812	5617	7423
0.9531	107	157	208	259	4428	8322	12216	16110
0.9919	617	911	1205	1499	25952	48423	70893	93364

7.7 Conclusions

In this chapter we have considered the Markovian polling systems, and have obtained the mean waiting times. It can be shown that the explicit expression for the expected waiting time of a customer conditioned on the system state at its arrival epoch has the linear functional form, which is the representative characteristic of our method. This form results from the linear functional forms of the basic quantities given in Proposition 7.1. And the steady-state average values can be derived from it by simple limiting procedures. It has been shown that the conditional expected waiting times in many types of M/G/1 multiclass queueing systems have the similar linear functional forms. They appear not only in the polling systems [2] but also in the priority systems [24]. Furthermore the conditional expected sojourn times in the systems with customers' feedback also have the linear functional forms [1], [25].

Our functional computation for the mean waiting times in the Markovian polling systems originally requires us to solve $J+1$ sets of $O(J^2)$ linear equations for the mean waiting times of J stations as opposed to the buffer occupancy method which requires us to solve $O(J^3)$ linear equations. Although our original method requires $O(J^7)$ numerical operations, we can construct the procedure with the successive approximation for the steady-state values which only requires $O(J^4)+O(J^3N)$ numerical operations where N is the number of its iterations. When we compared our method with the buffer occupancy method by actually computing the mean waiting times, we found that the computation times by our method are less than those by the buffer occupancy method; especially their differences are large when the system is congested.

Besides the above things, there are many advantage of our method [25]. Multiclass queueing models are useful for analyzing the computer communication systems with many datatypes and sources, and more complicated queueing models are necessary in order to derive the performance characteristics in the real systems. Because we can investigate complicated multiclass structures and composite scheduling algorithms by our method, it may stimulate advanced analysis of these systems.

Appendix: Proof of Proposition 7.2

Proof. We prove that the polling equation (7.8) is satisfied by directly substituting the expression for $\hat{H}_j(\mathbf{Y},e,l,k)$ defined by (7.28) into it. Let $\mathbf{Y}=(\kappa_0,r,\mathbf{g},\mathbf{n},L)\in\mathscr{E}$ be the state of the system at time τ_l^e $(l=0,1,\ldots,\ e=1,2,\ldots)$.

Case 1 ($k\in\Pi$): In the following expressions, the abbreviated condition $(\mathbf{Y},j)_l^e$ means the condition $\mathbf{Y}(\tau_l^e)=\mathbf{Y}$ and $X_S^e(\tau_l^e)=j$ for $l\geq 0,\ e=1,2,\ldots.$

For $(\kappa_0=j,l=0,j\in\mathscr{H}_e)$ or $(\kappa_0=j,l>0,j\in\mathscr{H}_e\cup\mathscr{H}_g)$, it can be easily shown that

$$\hat{H}_j(\mathbf{Y},e,l,k)=0.$$

For $(l=0,\kappa_0\in\Pi,\kappa_0\neq j)$ or $(l=0,\kappa_0=j\in\mathscr{H}_g)$,

$$\begin{aligned}
&H_j^0(\mathbf{Y},e,0,k)+\mathrm{E}[\hat{H}_j(\mathbf{Y}(\tau_1^e),e,1,k)|\mathbf{Y}(\tau_0^e)=\mathbf{Y},X_S^e(\tau_0^e)=j]\\
&=r\varphi^0(\kappa_0,j,k)+(\mathbf{g},\mathbf{n})\mathbf{h}_{00}^0(\kappa_0,j,k)\\
&\quad+\mathrm{E}[(\mathbf{g}(\tau_1^e),\mathbf{n}(\tau_1^e))\mathbf{h}_{10}(\kappa(\tau_1^e),j,k)+h_{11}(\kappa(\tau_1^e),j,k)|(\mathbf{Y},j)_0^e]\\
&=r\varphi^0(\kappa_0,j,k)+(\mathbf{g},\mathbf{n})\mathbf{h}_{00}^0(\kappa_0,j,k)+\sum_{\kappa_1\neq j}p_{\kappa_0\kappa_1}h_{11}(\kappa_1,j,k)\\
&\quad+\sum_{\kappa_1\neq j}p_{\kappa_0\kappa_1}\mathrm{E}[(\mathbf{g}(\tau_1^e),\mathbf{n}(\tau_1^e))|\kappa(\tau_1^e)=\kappa_1,(\mathbf{Y},j)_0^e]\mathbf{h}_{10}(\kappa_1,j,k)\\
&=r\varphi^0(\kappa_0,j,k)+(\mathbf{g},\mathbf{n})\mathbf{h}_{00}^0(\kappa_0,j,k)+\sum_{\kappa_1\neq j}p_{\kappa_0\kappa_1}h_{11}(\kappa_1,j,k)\\
&\quad+\sum_{\kappa_1\neq j}p_{\kappa_0\kappa_1}\left\{r\upsilon(\kappa_0)+(\mathbf{g},\mathbf{n})\mathbf{U}_0(\kappa_0)+\mathbf{u}_0(j,\kappa_0,\kappa_1)\right\}\mathbf{h}_{10}(\kappa_1,j,k)\\
&=\hat{H}_j(\mathbf{Y},e,0,k).
\end{aligned}$$

The first equation comes from (7.23) and (7.28). The second equation comes from the definition of the switching probability $p_{\kappa_0\kappa_1}$. The third equation comes from (7.25). The last equation comes from the definitions of the constants (in Sect. 7.4) and (7.28).

For $l = 0$ and $\kappa_0 = (k_0, k_1) \in \Pi^s$,

$$\begin{aligned}
&H_j^0(\mathbf{Y}, e, 0, k) + \mathrm{E}[\hat{H}_j(\mathbf{Y}(\tau_1^e), e, 1, k) | \mathbf{Y}(\tau_0^e) = \mathbf{Y}, X_S^e(\tau_0^e) = j] \\
&= \mathrm{E}[(\mathbf{g}(\tau_1^e), \mathbf{n}(\tau_1^e))\mathbf{h}_{10}(\kappa(\tau_1^e), j, k) + h_{11}(\kappa(\tau_1^e), j, k) | (\mathbf{Y}, j)_0^e] \\
&= \mathrm{E}[(\mathbf{g}(\tau_1^e), \mathbf{n}(\tau_1^e)) | \kappa(\tau_1^e) = k_1, (\mathbf{Y}, j)_0^e]\mathbf{h}_{10}(k_1, j, k) + h_{11}(k_1, j, k) \\
&= \left\{ r\upsilon + (\mathbf{g}, \mathbf{n})\mathbf{U}_0 + (\mathbf{0}, \mathbf{e}_j) \right\} \mathbf{h}_{10}(k_1, j, k) + h_{11}(k_1, j, k) \\
&= \hat{H}_j(\mathbf{Y}, e, 0, k).
\end{aligned}$$

For $l > 0$ and $\kappa_0 \neq j$ ($\kappa_0 \in \Pi$),

$$\begin{aligned}
&H_j^0(\mathbf{Y}, e, l, k) + \mathrm{E}[\hat{H}_j(\mathbf{Y}(\tau_{l+1}^e), e, l+1, k) | \mathbf{Y}(\tau_l^e) = \mathbf{Y}, X_S^e(\tau_l^e) = j] \\
&= (\mathbf{g}, \mathbf{n})\mathbf{h}_{10}^0(\kappa_0, j, k) \\
&\quad + \mathrm{E}[(\mathbf{g}(\tau_{l+1}^e), \mathbf{n}(\tau_{l+1}^e))\mathbf{h}_{10}(\kappa(\tau_{l+1}^e), j, k) + h_{11}(\kappa(\tau_{l+1}^e), j, k) | (\mathbf{Y}, j)_l^e] \\
&= (\mathbf{g}, \mathbf{n})\mathbf{h}_{10}^0(\kappa_0, j, k) + \sum_{\kappa_1 \neq j} p_{\kappa_0\kappa_1} h_{11}(\kappa_1, j, k) \\
&\quad + \sum_{\kappa_1 \neq j} p_{\kappa_0\kappa_1} \mathrm{E}[(\mathbf{g}(\tau_{l+1}^e), \mathbf{n}(\tau_{l+1}^e)) | \kappa(\tau_{l+1}^e) = \kappa_1, (\mathbf{Y}, j)_l^e]\mathbf{h}_{10}(\kappa_1, j, k) \\
&= (\mathbf{g}, \mathbf{n})\mathbf{h}_{10}^0(\kappa_0, j, k) + \sum_{\kappa_1 \neq j} p_{\kappa_0\kappa_1} h_{11}(\kappa_1, j, k) \\
&\quad + \sum_{\kappa_1 \neq j} p_{\kappa_0\kappa_1} \{(\mathbf{g}, \mathbf{n})\mathbf{U}_1(\kappa_0) + \mathbf{u}_1(\kappa_0, \kappa_1)\} \mathbf{h}_{10}(\kappa_1, j, k) \\
&= \hat{H}_j(\mathbf{Y}, e, l, k).
\end{aligned}$$

Case 2 ($k \in \Pi^s$): The proof is similar to case 1 and is omitted.
Hence the proof is completed. □

References

1. T. Hirayama, Mean sojourn times in multiclass feedback queues with gated disciplines, *Naval Research Logistics*, vol. 50, no. 7, pp. 719–741, 2003.
2. T. Hirayama, S. J. Hong, and M. M. Krunz, A new approach to analysis of polling systems, *Queueing Systems*, vol. 48, nos. 1–2, pp. 135–158, 2004.
3. M. M. Srinivasan, Nondeterministic polling systems, *Management Science*, vol. 37, no. 6, pp. 667–681, 1991.
4. H. Levy and M. Sidi, Polling systems: Applications, modeling, and optimization, *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1750–1760, 1990.
5. H. Takagi, Analysis and application of polling models, in: G. Haring, C. Lindemann, and M. Reiser (Eds.), *Performance Evaluation: Origins and Directions*, *Lecture Notes in Computer Science*, vol. 1769, pp. 423–442. Berlin: Springer, 2000.
6. R. B. Cooper, Queues served in cyclic order: Waiting times, *Bell System Technical Journal*, vol. 49, no. 3, pp. 399–413, 1970.

7. R. B. Cooper and G. Murray, Queues served in cyclic order, *Bell System Technical Journal*, vol. 48, no. 3, pp. 675–689, 1969.
8. M. Eisenberg, Queues with periodic service and changeover time, *Operations Research*, vol. 20, no. 2, pp. 440–451, 1972.
9. H. Takagi, Analysis of polling systems with a mixture of exhaustive and gated service disciplines, *Journal of the Operations Research Society of Japan*, vol. 32, no. 4, pp. 450–461, 1989.
10. H. Levy and M. Sidi, Polling systems with simultaneous arrivals, *IEEE Transactions on Communications*, vol. 39, no. 6, pp. 823–827, 1991.
11. M. Sidi, H. Levy and S. W. Fuhrmann, A queueing network with a single cyclically roving server, *Queueing Systems*, vol. 11, nos. 1–2, pp. 121–144, 1992.
12. M. J. Ferguson and Y. J. Aminetzah, Exact results for nonsymmetric token ring systems, *IEEE Transactions on Communications*, vol. 33, no. 3, pp. 223–231, 1985.
13. D. Sarkar and W. I. Zangwill, Expected waiting time for nonsymmetric cyclic queueing systems — Exact results and applications, *Management Science*, vol. 35, no. 12, pp. 1463–1474, 1989.
14. E. M. M. Winands, I. J. B. F. Adan, and G. J. Van Houtum, Mean value analysis for polling systems, *Queueing Systems*, vol. 54, no. 1, pp. 35–44, 2006.
15. H. Takagi, *Analysis of Polling Systems*. Cambridge: MIT Press, 1986.
16. A. G. Konheim, H. Levy, and M. M. Srinivasan, Descendant set: An efficient approach for the analysis of polling systems, *IEEE Transactions on Communications*, vol. 42, nos. 2/3/4, pp. 1245–1253, 1994.
17. M. P. Singh and M. M. Srinivasan, Exact analysis of the state-dependent polling model, *Queueing Systems*, vol. 41, no. 4, pp. 371–399, 2002.
18. O. J. Boxma, Workloads and waiting times in single-server systems with multiple customer classes, *Queueing Systems*, vol. 5, nos. 1–3, pp. 185–214, 1989.
19. R. B. Cooper, S-C. Niu, and M. M. Srinivasan, A decomposition theorem for polling models: The switchover times are effectively additive, *Operations Research*, vol. 44, no. 4, pp. 629–633, 1996.
20. L. Kleinrock and H. Levy, The analysis of random polling systems, *Operations Research*, vol. 36, no. 5, pp. 716–732, 1988.
21. H. Chung, C. K. Un, and W. Y. Jung, Performance analysis of Markovian polling systems with single buffers, *Performance Evaluation*, vol. 19, no. 4, pp. 303–315, 1994.
22. J. E. Baker and I. Rubin, Polling with a general-service order table, *IEEE Transactions on Communications*, vol. 35, no. 3, pp. 283–288, 1987.
23. O. J. Boxma, H. Levy, and J. A. Weststrate, Efficient visit frequencies for polling tables: Minimization of waiting cost, *Queueing Systems*, vol. 9, nos. 1–2, pp. 133–162, 1991.
24. T. Hirayama, Analysis of multiclass M/G/1 queues with a mixture of 1-limited disciplines and gated disciplines, *Journal of the Operations Research Society of Japan*, vol. 42, no. 3, pp. 237–255, 1999.
25. T. Hirayama, Multiclass polling systems with Markovian feedback: Mean sojourn times in gated and exhaustive systems with local priority and FCFS service orders, *Journal of the Operations Research Society of Japan*, vol. 48, no. 3, pp. 226–255, 2005.
26. H. Levy, Delay computation and dynamic behavior of non-symmetric polling systems, *Performance Evaluation*, vol. 10, no. 1, pp. 35–51, 1989.
27. W. Whitt, A review of $L = \lambda W$ and extensions, *Queueing Systems*, vol. 9, no. 3, pp. 235–268, 1991.