

Control Policies for the $M^X/G/1$ Queueing System

Hyo-Seong Lee

M. M. Srinivasan

Department of Industrial and Operations Engineering

The University of Michigan, Ann Arbor 48109-2117

Technical Report 87-20

September 1987

Abstract

The $M^x/G/1$ queueing system is studied under the following two situations:

- (1) At the end of a busy period, the server is turned off and inspects the length of the queue every time an arrival occurs. When the queue length reaches, or exceeds, a pre-specified value m for the first time, the server is turned on and serves the system until it is empty.
- (2) At the end of a busy period, the server takes a sequence of vacations, each for a random amount of time. At the end of each vacation, he inspects the length of the queue. If the queue length is greater than, or equal to, a pre-specified value m at this time, he begins to serve the system until it is empty.

For both cases, the mean waiting time of an arbitrary customer for a given value of m is derived, and the procedure to find the stationary optimal policy under a linear cost structure is presented.

1. Introduction

In many real queueing systems, requests for service usually arrive in batches. For example, in manufacturing systems of the job-shop type, each job order often requires the manufacture of more than one unit; in computer communication systems, messages which are to be transmitted could consist of a random number of packets. When a cost structure is superimposed on such batch arrival systems, obtaining an optimal control policy under the given cost structure is very important. However, while a considerable amount of work has been done on control policies for queueing systems without batch arrivals, comparable work for the batch arrival case is rarely found in the literature. This motivates us to develop control policies for queueing systems with batch arrivals.

We consider an $M^*/G/1$ queueing system, where requests for service arrive to the system according to a Poisson process. Each request consists of a number, X , of units where X is an independent and identically distributed random variable. The units within a batch are served one at a time by a single server. The service time for an individual unit is independent and identically distributed. In this paper, we consider control policies for the following two situations.

- (a) Every time an arrival occurs, the server inspects the state of the queue. If the total number of units is found to have reached or exceeded a pre-specified value m , he takes a random amount of time to start up the system. Following this, the server begins to serve the queue until the system is empty. As soon as the system becomes empty, the server

is turned off and remains idle until the accumulated number of units reaches or exceeds m for the first time.

- (b) In this case, when the server finishes service and finds the system empty, he takes a vacation for a random amount of time. At the end of the vacation, the server determines the number of units present in the queue. If the queue length at that time is below m , he immediately takes another vacation. On the other hand, if the queue length is greater than, or equal to m , he takes a random amount of time to start up the system. He then begins to serve the system until it is empty.

It is to be noted that while the queueing system in (a) is continuously monitored, the system in (b) is monitored only at the times each vacation ends. In this paper, we refer to the policy in (a) as the m-policy without vacation and the policy in (b) as the m-policy with vacation.

In these systems, it is assumed that there is a cost incurred each time the server is turned on, and that there is a cost incurred for every unit of time a unit waits for service. For both these policies, we obtain the mean waiting time of an arbitrary unit for a given value of m and find the stationary optimal m -policy which minimizes the expected cost in the long run, under this cost structure.

2. Previous Work

As mentioned earlier, a considerable amount of work has been done on control policies for the $M/G/1$ queueing system without batch arrivals. Most of these studies present some special policies and derive the optimal

stationary policies under certain cost structures. Among them, the N-policy studied by Yadin and Naor[17], Heyman[8], Sobel[16] and Bell[4] is one of the earliest. In this policy, the server is turned on when the total number of units reaches N and turned off when the system becomes empty.

Heyman[9] also considered the T-policy where the server is turned off at the end of each busy period and inspects the state of the queue after an interval of length T . If at least one unit is found awaiting service, a busy period begins and the units are served until the queue is exhausted. If there are no units found at the time of inspection, the server again inspects the queue after another interval of length T . Heyman showed that under a linear cost structure the optimal N-policy is superior to the optimal T-policy.

Balachandran[2] and Balachandran and Tijms[3] proposed the D-policy in which the server is turned off at the end of a busy period and turned back on when the cumulative service time of the waiting units exceeds some fixed value D . They also showed[3] that under a linear cost structure the D-policy is superior to the N-policy for the service time distributions with decreasing, constant, and for some cases, increasing failure rates.

Hofri[10] studied the control policy for the M/G/1 system in which the server takes a sequence of vacations at the end of busy period, inspecting the state of the queue at the end of each vacation. When the queue length exceeds a predetermined level m , the units are served until the system

becomes empty. Furthermore, he considered a control policy for the system where two queues are served by a single server.

It is important to note that all these studies fall under the class of server vacation models for the M/G/1 queueing system, which have been extensively studied by Levy and Yechiali[12], Fuhrmann and Cooper[7], Scholl and Kleinrock[15], and Levy and Kleinrock[11], to name but a few.

The m-policy in this paper extends the work on control policies to the situation where arrivals occur according to a discrete compound Poisson process and a start-up interval is allowed which takes a random amount of time. When the batch size is 1 and the start-up interval is zero, our m-policy without vacation reduces to the N-policy model of Heyman, and our m-policy with vacation reduces to the vacation model studied by Hofri.

In the case of batch arrivals, it is also possible to adopt a policy that uses information on the number of batches, instead of the total number of individual units in deciding whether the server is activated or not. This policy, which we will term as the n-policy in this paper, is described as follows:

(1) n-policy without vacation

At the end of a busy period, the server is turned off and begins to count the arrivals as and when they occur. When the total number of arrivals reaches a pre-specified value n , the server takes a random amount of time to start up the system. He then begins to serve the queue exhaustively.

(2) n-policy with vacation

At the end of a busy period, the server takes a sequence of vacations. At the end of each vacation, the server counts the number of batches that have queued up for service. If the total number of batches at this time equals to or exceeds a pre-specified value n , the server takes a random amount of time to start up the system, and then begins to serve the queue exhaustively. Otherwise, the server immediately takes another vacation.

If the n-policy is adopted, the mean waiting time of an arbitrary unit for a given value of n is obtained by treating a batch as if it were a single unit, i.e., (1) find the mean waiting time of a typical batch and (2) add to it the mean waiting time of a test(arbitrary) unit within a batch. This technique, however, cannot be applied to the m-policy since the m-policy uses the information about the individual units instead of batches. Therefore, known results by previous authors cannot be directly applied to the m-policy. This fact has led us to devise a different approach to obtain the mean waiting time experienced by an arbitrary unit in the m-policy. In section 7, the m-policy and the n-policy will be compared and it will be shown that the optimal m-policy is superior to the optimal n-policy under a linear cost structure.

3. The Approach

Requests arrive in batches according to a stationary Poisson stream with rate λ . Let X denote the number of units in a batch, where X is a random variable (r.v.) with a known discrete distribution. The service time

of a typical unit is denoted by S and we assume that S has a finite second moment.

Since we are only interested in the first moment of the waiting time, the service discipline is not a matter of concern to us. However, for analytical convenience, we assume that service is done in a First In First Out (FIFO) manner.

We define some terms which will be used in the analysis.

Primary units are defined to be units which arrive either while the server is turned off, or is being started up, and secondary units are those which arrive while the server is busy. We define a cycle to be the period that elapses from the end of a busy period till the end of the next busy period. In order to analyze the system, we divide the cycle into the following three phases.

Phase 1 begins when the server is turned off and lasts until the server has completed the start-up. Thus, phase 1 consists of two sub-periods: a dormant period during which the server is turned off and a start-up period during which the server is starting up the system. Note that the units which arrive during phase 1 are the primary units.

Phase 2 begins when the start-up by the server is completed, and lasts until all the primary units are served.

Phase 3 begins at the end of phase 2 and lasts until all the secondary units are served.

Note that primary units arrive during phase 1 and are served during phase 2. Secondary units arrive during both phase 2 and phase 3, and are served during phase 3.

Our general approach to obtain the mean waiting time experienced by an arbitrary unit in the m -policy is to analyze the mean waiting times of the primary and the secondary units separately, and then to combine these results to yield the mean waiting time of an arbitrary unit. To this end, let W_m^p (W_m^s) be the mean waiting time of a primary (secondary) unit, and let N_m^p (N_m^s) be the mean number of primary (secondary) units served in a cycle when we use m as a control value. Then the mean waiting time of an arbitrary unit, W_m , is obtained as

$$W_m = \frac{N_m^p W_m^p + N_m^s W_m^s}{N_m^p + N_m^s}. \quad (3.1)$$

To obtain W_m^p in (3.1), we divide W_m^p into two components, $W_m^{(p,1)}$ and $W_m^{(p,2)}$, where $W_m^{(p,1)}$ is the mean waiting time of an arbitrary primary unit in phase 1 and $W_m^{(p,2)}$ is its mean waiting time in phase 2. We obtain $W_m^{(p,1)}$ and $W_m^{(p,2)}$ in isolation and obtain W_m^p as:

$$W_m^p = W_m^{(p,1)} + W_m^{(p,2)}. \quad (3.2)$$

In the following sections, we develop the expressions for the terms $W_m^{(p,1)}$, $W_m^{(p,2)}$, W_m^s , N_m^p , N_m^s and the procedure to find the optimal m -policies with/without vacations.

4. m-Policy with Vacation

In this section, we obtain the mean waiting time of an arbitrary unit in a system using the m-policy with vacation. In the m-policy with vacation, as explained in section 1, the server takes a sequence of vacations at the end of a busy period until the queue length reaches or exceeds m for the first time, at which point the start-up period begins immediately; as soon as the start-up is finished the server begins to serve the queue. For any discrete random variable, A, used in the analysis, we adopt the following notation:

$$a_i = P\{A=i\},$$

$$A(z) = \sum_{i=0}^{\infty} a_i z^i, \quad \text{the probability generating function (p.g.f.) of A,}$$

$$a^{(1)} = E(A),$$

$$a^{(2)} = E(A(A-1)).$$

Similarly, for any continuous random variable B used in the analysis, B(.) and B*(.) will denote the distribution function and Laplace-Stieltjes Transform(LST) of B respectively.

We define

$$V = \text{vacation time, r.v.,}$$

$$Q = \text{number of batches that arrive during a vacation, r.v.,}$$

$$U = \text{start-up time, r.v.,}$$

$$R = \text{number of units that arrive during a vacation, r.v.,}$$

$$I_{d,m} = \text{number of (primary) units that arrive during a dormant period,} \\ \text{when m is used as a control value, r.v.,}$$

$$I_s = \text{number of (primary) units arriving during a start-up period, r.v.,}$$

$$I_m = I_{d,m} + I_s = \text{number of (primary) units present when the service} \\ \text{begins, when m is used as a control value, r.v.,}$$

$L_{d,m}$ = expected total waiting time of primary units during a dormant period, when m is used as a control value,

$L_{s,m}$ = expected total waiting time of primary units during a start-up period, when m is used as a control value,

$L_m = L_{d,m} + L_{s,m}$ = expected total waiting time of all the primary units in phase 1, when m is used as a control value,

$\ell_{d,m}$ = expected length of dormant period, when m is used as a control value,

$\ell_m = \ell_{d,m} + E(U)$ = expected length of phase 1, when m is used as a control value,

$x_j^{*(i)} = P\{i\text{-fold convolution of } X \text{ is } j\}.$

From these definitions, the following relationships are obtained:

a) By definition,

$$q_i = \int_0^{\infty} \frac{(\lambda t)^i}{i!} e^{-\lambda t} dV(t). \quad (4.1)$$

However, for the computation of q_i , we use the following equivalent expression:

$$q_i = \frac{(-\lambda)^i}{i!} V^{*(i)}(\lambda), \quad (4.2)$$

where $V^{*(i)}(\theta)$ is the i^{th} derivative of the LST of V with respect to θ .

b) The probability that j units arrive during a vacation is given by

$$r_j = \sum_{i=0}^j q_i x_j^{*(i)}, \quad j=0,1,2,\dots. \quad (4.3)$$

We also present a lemma that will be useful for computing the moments of R , and for obtaining the pgf of $I_{d,m}$:

Lemma 4.1. The probability generating function of R is given by

$$R(z) = V^*(\lambda - \lambda X(z)). \quad (4.4.a)$$

$$I_s(z) = U^*(\lambda - \lambda X(z)). \quad (4.4.b)$$

(proof) By definition of the p.g.f.,

$$\begin{aligned} R(z) &= \sum_{j=0}^{\infty} r_j z^j \\ &= \sum_{j=0}^{\infty} \sum_{i=0}^j q_i x_j^{*(i)} z^j \\ &= \sum_{i=0}^{\infty} q_i \sum_{j=i}^{\infty} x_j^{*(i)} z^j \\ &= \sum_{i=0}^{\infty} q_i \{X(z)\}^i \\ &= Q(X(z)). \end{aligned}$$

Noting that $Q(z) = V^*(\lambda - \lambda z)$, the result follows.

Equation (4.4b) is proved in a similar manner. ■

From lemma 4.1, by differentiating $R(z)$ and $I_s(z)$ with respect to z , we can obtain the first and second factorial moments of R and I_s as:

$$r^{(1)} = E(R) = \lambda E(V)_X^{(1)}, \quad (4.5.a)$$

$$r^{(2)} = E(R(R-1)) = (\lambda x^{(1)})^2 E(V^2) + \lambda x^{(2)} E(V), \quad (4.5.b)$$

$$i_s^{(1)} = E(I_s) = \lambda E(U)_X^{(1)}, \quad (4.5.c)$$

$$i_s^{(2)} = E(I_s(I_s-1)) = (\lambda x^{(1)})^2 E(U^2) + \lambda x^{(2)} E(U). \quad (4.5.d)$$

4.1 Computing the term $W_m^{(p,1)}$:

Suppose we have available the values of L_m and $i_m^{(1)}$. Then, the mean waiting time of a primary test(arbitrary) unit in phase 1 is obtained as

$$W_m^{(p,1)} = \frac{L_m}{i_m^{(1)}}. \quad (4.6)$$

The term $i_m^{(1)}$ consists of two terms, $i_{d,m}^{(1)}$ and $i_s^{(1)}$. In order to obtain $i_{d,m}^{(1)}$,

we need to find the p.g.f. of $I_{d,m}$. The following result, stated as Theorem 4.1 expresses the p.g.f. of $I_{d,m}$. A proof of Theorem 4.1 is given in the Appendix.

Theorem 4.1.

$$I_{d,m}(z) = \sum_{j=1}^{m-1} \frac{r_j}{1-r_0} z^j (I_{d,m-j}(z)-1) + \frac{R(z)-r_0}{1-r_0}, \quad m=1,2,\dots \quad (4.7)$$

■

Using $I_{d,m}(z)$ we obtain the first and second factorial moments of $I_{d,m}$. The second moment of $I_{d,m}$ will be used to calculate $i_m^{(2)}$.

$$i_{d,m}^{(1)} = E(I_{d,m}) = \frac{1}{1-r_0} \sum_{j=1}^{m-1} r_j i_{d,m-j}^{(1)} + \frac{r^{(1)}}{1-r_0}, \quad m=1,2,\dots \quad (4.8.a)$$

$$i_{d,m}^{(2)} = E(I_{d,m}(I_{d,m}-1)) = \frac{1}{1-r_0} \sum_{j=1}^{m-1} r_j (2j i_{d,m-j}^{(1)} + i_{d,m-j}^{(2)}) + \frac{r^{(2)}}{1-r_0}, \quad m=1,2,\dots \quad (4.8.b)$$

Since the number of units which arrive during the start-up period is independent of the number of units which arrive during the dormant period, the following relationship holds:

$$I_m(z) = I_{d,m}(z)I_s(z). \quad (4.9)$$

By differentiating $I_m(z)$ with respect to z , we obtain the first and second factorial moments of $I_m(z)$ as:

$$i_m^{(1)} = i_{d,m}^{(1)} + i_s^{(1)}, \quad (4.10.a)$$

$$i_m^{(2)} = i_{d,m}^{(2)} + i_s^{(2)} + 2i_{d,m}^{(1)} i_s^{(1)}, \quad (4.10.b)$$

where $i_s^{(1)}$ and $i_s^{(2)}$ are obtained from equations (4.5.c) and (4.5.d).

Now the term $i_m^{(1)}$ can be obtained from (4.10.a). To compute the term L_m which is needed in (4.6), we first obtain the term $L_{d,m}$ and then obtain the term $L_{s,m}$. In order to compute the term $L_{d,m}$, we require the following lemma.

Lemma 4.2: Consider all the units that arrive during the first vacation. Let ω denote the total waiting time experienced by these units during this vacation. Then its expected value, $E(\omega)$ is given by

$$E(\omega) = \frac{1}{2} x^{(1)} \lambda E(V^2). \quad (4.11)$$

(proof) Lemma 4.2 is proved using the following simple argument.

- 1) If we take an arbitrary unit which has arrived during the first vacation, its mean waiting time during the first vacation is the expected residual life time of a vacation, i.e., $\frac{E(V^2)}{2E(V)}$.
- 2) The expected number of units which arrive during the first vacation is $\lambda E(V)x^{(1)}$.

The product of terms 1) and 2) yields $E(\omega) = \frac{1}{2} x^{(1)} \lambda E(V^2)$.

■

The term, $L_{d,m}$, is then obtained from the following recursive relation:

$$L_{d,m} = E(\omega) + \sum_{j=0}^{m-1} r_j (L_{d,m-j} + L_{d,m-j}), \quad m=1,2,\dots \quad (4.12)$$

The explanation of equation (4.12) is as follows:

For any value of m , the m -policy must have at least one vacation during the dormant period. Therefore, $L_{d,m}$ is represented by the expected total

waiting time during the first vacation plus the expected total waiting time from the second vacation onwards until the end of dormant period. In (4.12), the first term expresses the expected total waiting time during the first vacation for all the units that arrive during this vacation and the second term expresses the expected total waiting time from the second vacation onward until the end of dormant period. The expression for the second term is based on the following reasoning:

- 1) If the number of units that arrive during the first vacation is more than or equal to m , dormant period ends as soon as the first vacation finishes and there is no further waiting in dormant period.
- 2) Suppose j units arrived during the first vacation. The probability that j units arrive during the first vacation is r_j . If j is less than m , each of these j units must wait for $l_{d,m-j}$ on the average from the beginning of the second vacation onward until the end of dormant period. Noting that the expected total waiting time of the units that arrive from the second vacation onwards is $L_{d,m-j}$ we get the second term of equation (4.12).

If we solve equation (4.12) for $L_{d,m}$ we obtain

$$L_{d,m} = \frac{1}{1-r_0} \left\{ \frac{1}{2} x^{(1)} \lambda E(V^2) + \sum_{j=1}^{m-1} r_j (j l_{d,m-j} + L_{d,m-j}) \right\}, \quad m=1,2,\dots, \quad (4.13)$$

where $\sum_{j=1}^0$ is defined to be 0.

The term, $l_{d,k}$, which is needed (for $k=1,2,\dots,m-1$) in equation (4.13), can be obtained by the following forward-type recursive equation.

$$l_{d,k} = E(V) + \sum_{j=0}^{k-1} r_j l_{d,k-j}, \quad k=1,2,\dots. \quad (4.14)$$

The first term of equation (4.14) represents the fact that at least one vacation is needed for any value of m used in the m -policy. The second term of equation (4.14) is obtained using the fact that if the number of units which arrive during the first vacation is less than k , say j , the expected remaining length of dormant period is $l_{d,k-j}$, and also the probability that j units arrive during the first vacation is r_j . If we solve equation (4.14) for $l_{d,k}$, we obtain

$$l_{d,k} = \frac{1}{1-r_0} \{E(V) + \sum_{j=1}^{k-1} r_j l_{d,k-j}\}, \quad k=1,2,\dots \quad (4.15)$$

where, again, $\sum_{j=1}^0$ is defined to be 0.

Now that we can calculate the term $L_{d,m}$, the only remaining unknown needed to obtain $W_m^{(p,1)}$ is $L_{s,m}$, which is obtained by the following equation.

$$L_{s,m} = i_{d,m}^{(1)} E(U) + \frac{\lambda x^{(1)} E(U^2)}{2}. \quad (4.16)$$

In equation (4.16), the first term represents the expected total waiting time of the units during the start-up period which are already present at the beginning of that period. The second term represents the expected total waiting time for the units which arrive during that period. The second term is obtained using an approach similar to that used to obtain equation (4.11).

Now L_m is determined by summing $L_{d,m}$ and $L_{s,m}$, and $i_m^{(1)}$ is obtained from equation (4.10.a). Thus we can finally calculate $W_m^{(p,1)}$ using equation (4.6).

Remark 1 From a computational point of view, q_i is easily obtained from equation (4.2) if $V^*(.)$ is a well defined differentiable function. On the other hand, we may have trouble calculating r_j if j is large since calculation of $g_j^{*(i)}$ is usually cumbersome if i is large (refer equation (4.3)). However, from a practical point of view, we can always calculate r_j to the desired level of accuracy for the following reasons:

- 1) If the value of m used in the m -policy is not so big, we can calculate all the necessary r_j s without difficulty since only r_0, r_1, \dots, r_{m-1} are needed to obtain $W_m^{(p,1)}$.
- 2) Considering that q_i converges to 0 very fast and $x_j^{*(i)}$ is always less than or equal to 1, we can calculate r_j using equation (4.3) by truncating the summation after k terms are summed up if $q_k \approx 0$ for some $k < j$.

4.2. Determining $W_m^{(p,2)}$:

We are now in a position to find $W_m^{(p,2)}$, the mean waiting time of the primary test unit in phase 2. Let I_m' be the number of primary units in a cycle that contains the test unit. Then, $W_m^{(p,2)}$ is obtained by the following equation (see, for example, Cooper[6]):

$$W_m^{(p,2)} = \sum_{j=m}^{\infty} E(W_m^{(p,2)} \mid I_m' = j) P\{I_m' = j\}, \quad (4.17)$$

where

$$P\{I_m' = j\} = \frac{j P\{I_m = j\}}{i_m^{(1)}}, \quad (4.18)$$

and

$$E(W_m^{(p,2)} | I_m' = j) = \frac{(j-1)}{2} E(S). \quad (4.19)$$

Substitution of (4.18) and (4.19) into (4.17) yields

$$W_m^{(p,2)} = \frac{i_m^{(2)} E(S)}{2 i_m^{(1)}}. \quad (4.20)$$

4.3. Determining W_m^s :

To determine the mean waiting time for secondary units it is first noted that their waiting time in the system occurs only during phases 2 and 3. Further, note that as far as these units are concerned, phase 2 represents a 'vacation' period during which the arriving secondary units find the server busy attending to primary units. Phase 3 then represents a busy period for these secondary units. Let T_m represent the duration of phase 2, which is the pseudo-vacation (for the secondary units). We first determine the first and second moments of T_m . For this, note that T_m is a sum of the service times of the primary units, and can be expressed as:

$$T_m = S_1 + S_2 + \dots + S_{I_m}.$$

The LST of T_m is then given by

$$T_m^*(\theta) = \sum_{k=m}^{\infty} \{S^*(\theta)\}^k P\{I_m = k\}$$

$$= I_m(S^*(\theta)). \quad (4.21)$$

By differentiating (4.21), we obtain

$$E(T_m) = i_m^{(1)} E(S), \quad (4.22)$$

$$E(T_m^2) = i_m^{(2)} (E(S))^2 + i_m^{(1)} E(S^2). \quad (4.23)$$

Having determined the first and second moments of T_m , we can now obtain W_m^s using Lemma 4.3.

Lemma 4.3. Consider an $M^x/G/1$ server vacation model where the server begins a vacation of random length T each time the system becomes empty. If the server finds at least one unit at the end of a vacation, he begins to serve the units until the queue is empty. If the server finds no units waiting at the end of the vacation, he takes another vacation immediately. The arrival rate, batch size and service time for an individual unit are assumed to follow our general notations. Then, the mean waiting time, W , for an arbitrary unit is given by

$$W = \frac{E(T^2)}{2E(T)} + \frac{\lambda(x^{(1)}E(S^2) + x^{(2)}(E(S))^2)}{2(1-\rho)} + \frac{E(S)x^{(2)}}{2x^{(1)}}, \quad (4.24)$$

where $\rho = \lambda x^{(1)} E(S)$. ■

Remark 2 Baba obtained equation (4.24) using a supplementary variable technique in [1]. However, equation (4.24) can be obtained directly if we combine the decomposition property of Furmann and Cooper[7] for the $M/G/1$ system with server vacation and Burke's method[5] to obtain the expected waiting time for the $M^x/G/1$ system. That is, we first treat a batch as if it were an individual unit. Then by the decomposition property of the

M/G/1 server vacation model, the mean waiting time for a typical batch is expressed as the sum of the mean residual life time of a vacation plus the mean waiting time of a typical batch in the corresponding standard M/G/1 system. Then, using Burke's method, the mean waiting time of a test unit is obtained by adding the mean waiting time of a test unit within a batch to the mean waiting time for a typical batch. Therefore W consists of the following three independent components: the mean residual life time of a vacation, the mean waiting time of a batch in the corresponding M/G/1 system and the mean waiting time of the test unit within a batch. Each of these three components is represented by the first, second and the third terms of equation (4.24) respectively. ■

Substituting T_m in place of T in Lemma (4.3), and using equations (4.22) and (4.23), we obtain W_m^s as:

$$W_m^s = \frac{E(T_m^2)}{2E(T_m)} + \frac{\lambda\{x^{(1)}E(S^2) + x^{(2)}(E(S))^2\}}{2(1-\rho)} + \frac{E(S)x^{(2)}}{2x^{(1)}} ,$$

which can be rewritten, after some elementary algebra, as

$$W_m^s = \frac{i_m^{(2)}}{2i_m^{(1)}} E(S) + \frac{\lambda\{x^{(1)}E(S^2) + x^{(2)}(E(S))^2\}}{2\rho(1-\rho)} . \quad (4.25)$$

4.4. Determining N_m^p and N_m^s :

It is clear that,

$$N_m^p = i_m^{(1)} . \quad (4.26)$$

To obtain N_m^s , we again treat a batch as if it were a single unit. Let Y be the number of batches that arrive during phase 2. Then, the length of phase 3 becomes a sum of Y independent busy periods of an M/G/1 queue. Applying the well known fact about the number of units served during the busy period of the M/G/1 queue, the expected number of batches served during phase 3 is $E(Y)/(1-\rho)$ where $E(Y)=\lambda E(T_m)$. Since N_m^s is the expected number of units served during phase 3, it is expressed as

$$N_m^s = \frac{\lambda E(T_m) x^{(1)}}{1-\rho},$$

and using equation (4.22), this simplifies to

$$N_m^s = \frac{\rho}{1-\rho} i_m^{(1)}. \quad (4.27)$$

From equations (4.26) and (4.27), equation (3.1) reduces to

$$W_m = (1-\rho)W_m^p + \rho W_m^s. \quad (4.28)$$

Finally from equations (4.6), (4.20), (4.25) and (4.28), we obtain an expression for W_m , stated as Theorem 4.2:

Theorem 4.2:

The mean waiting time, W_m , for the m-Policy $M^X/G/1$ queueing system with vacations is given by

$$W_m = (1-\rho) \frac{L_m}{i_m^{(1)}} + \frac{i_m^{(2)}}{2i_m^{(1)}} E(S) + \frac{\lambda \{x^{(1)} E(S^2) + x^{(2)} (E(S))^2\}}{2(1-\rho)} \quad (4.29)$$

■

Remark 3 As we see in equation (4.29), W_m consists of three terms which represent, respectively, the mean waiting time of a primary unit in phase 1 multiplied by $1-\rho$, the mean waiting time of a primary unit in phase 2, and the mean waiting time of a batch in the corresponding M/G/1 system. Since the fraction of primary units is $1-\rho$, the sum of the second and third terms of (4.29) represents the mean waiting time of an arbitrary unit after service begins, which is completely determined by the first and second moments of the number of primary units. ■

Let the busy period, which is the sum of phase 2 and phase 3, be denoted by B_m and let the LST of B_m be denoted by $B_m^*(\theta)$. It has been shown [1,13] that LST of the busy period initiated with 1 unit in the system, $B^*(\theta)$, can be expressed by

$$B^*(\theta) = S^*(\theta + \lambda - \lambda X(B^*(\theta))) , \quad (4.30)$$

and that the LST of the busy period initiated with k units in the system is expressed by $(B^*(\theta))^k$. Therefore, the LST of the busy period in the m -policy is

$$\begin{aligned} B_m^*(\theta) &= \sum_{j=m}^{\infty} P\{I_m=j\} (B^*(\theta))^j \\ &= I_m(B^*(\theta)) . \end{aligned} \quad (4.31)$$

From (4.30), we obtain the mean busy period initiated with 1 unit in the system as

$$E(B) = \frac{E(S)}{(1-\rho)} . \quad (4.32)$$

By differentiating (4.31) with respect to θ and using (4.32), we obtain the expected length of a busy period in the m -policy as

$$E(B_m) = \frac{i_m^{(1)} E(S)}{1-\rho} . \quad (4.33)$$

It can easily be shown (refer Property 4 in Section 6) that $\ell_m = \frac{i_m^{(1)}}{\lambda_X(1)}$. Using

this result, the mean length of a cycle in the m-policy is then expressed as

$$\ell_m + E(B_m) = \frac{i_m^{(1)}}{(1-\rho)\lambda_X(1)}. \quad (4.34)$$

Let the number of units served in a busy period initiated with 1 unit in the system be M and let the pgf of M be $M(z)$. Then it is easy to show that

$$M(z) = z S^*(\lambda - \lambda X(M(z))) \quad (4.35)$$

If we denote the number of units served in a busy period in the m-policy by M_m and its pgf by $M_m(z)$, $M_m(z)$ can be expressed as

$$\begin{aligned} M_m(z) &= \sum_{j=m}^{\infty} P(I_m=j) (M_m(z))^j \\ &= I_m(M(z)). \end{aligned} \quad (4.36)$$

By differentiating (4.36) with respect to z , we obtain the mean number of units served in a busy period (or in a cycle), N_m , as:

$$N_m = \frac{i_m^{(1)}}{1-\rho}. \quad (4.37)$$

Note that equation (4.37) can be also obtained from equations (4.26) and (4.27), using the fact that $N_m = N_m^p + N_m^s$.

5. m-Policy without Vacation

The general approach used to obtain W_m for the case of the m-policy without vacation is basically similar to that outlined in the case of the m-policy with vacation. Most of the equations used in the m-policy with vacation still hold and are used in the exactly same way. However, the

expressions for $I_{d,m}(z)$, $i_{d,m}^{(1)}$, $i_{d,m}^{(2)}$, $L_{d,m}$ and $\ell_{d,m}$ are different from those in the m-policy with vacation model. In this section, we obtain these alternate expressions.

The probability generating function of I_m , which yields $i_{d,m}^{(1)}$ and $i_{d,m}^{(2)}$, is expressed by the following theorem.

Theorem 5.1

$$I_{d,m}(z) = \sum_{j=1}^{m-1} x_j z^j \{I_{d,m-j}(z) - 1\} + X(z), \quad m=1,2,\dots \quad (5.1)$$

A proof of Theorem 5.1 is given in the Appendix.

From (5.1), the first and second factorial moments of $I_{d,m}$ are obtained as

$$i_{d,m}^{(1)} = \sum_{j=1}^{m-1} x_j i_{d,m-j}^{(1)} + x^{(1)}, \quad m=1,2,\dots, \quad (5.2)$$

$$i_{d,m}^{(2)} = \sum_{j=1}^{m-1} x_j (2j i_{d,m-j}^{(1)} + i_{d,m-j}^{(2)}) + x^{(2)}, \quad m=1,2,\dots \quad (5.3)$$

The expected total waiting time in dormant period, $L_{d,m}$, is obtained by conditioning on the first arrival. Suppose the number of units in the first batch is j . If j exceeds m , dormant period ends upon the occurrence of the first arrival so that there is no waiting in dormant period. If j is less than m , these j units must wait for the length of $\ell_{d,m-j}$ on the average and the expected total waiting time of the (primary) units, excluding these j units, is $L_{d,m-j}$. The term $L_{d,m}$ can be obtained, therefore, by the following recursive equation

$$L_{d,m} = \sum_{j=1}^{m-1} x_j (j \ell_{d,m-j} + L_{d,m-j}), \quad m=1,2,\dots \quad (5.4)$$

where $L_{d,1}=0$ and $\sum_{j=1}^0$ is defined to be 0.

The terms $\ell_{d,k}$, for $k=1,\dots,m-1$, in equation (5.4), is obtained from

$$\ell_{d,k} = \frac{1}{\lambda} + \sum_{j=1}^{k-1} x_j \ell_{d,k-j}, \quad k=1,2,\dots \quad (5.5)$$

It is easy to see that we can obtain equation (5.5) by a similar argument as

was used to obtain equation (4.14). To obtain $i_s^{(1)}, i_s^{(2)}, i_m^{(1)}, i_m^{(2)}, L_{s,m}$ and L_m

we use the same equations as in the m-policy with vacation. If we substitute $i_m^{(1)}$ and L_m into (4.6), we obtain $W_m^{(p,1)}$. To obtain $W_m^{(p,2)}, W_m^s, N_m^p$ and N_m^s , we

just follow the same procedure as described in section 4 using, however, the new expressions for $L_m, i_m^{(1)}$ and $i_m^{(2)}$ derived above.

6. Optimal Design of the m-Policy

In this section, we find the value of m which produces the optimal stationary m-policy. We assume that a start-up cost of c_s is incurred each time the server is turned on and that a linear holding cost of c_h per unit time is incurred for each unit that waits in this system. To obtain the optimal m policy, we first obtain the expected cost per unit for a given value of m , and then find the value of m which minimizes this cost.

Since the expected number of units served during a cycle is N_m , the expected start-up cost per unit is expressed as $\frac{c_s}{N_m}$. Similarly, the expected

waiting cost per unit is given by $c_h W_m$. From this, the expected (long run) total cost per unit, C_m , using m as a control value, is calculated as

$$\begin{aligned} C_m &= \frac{c_s}{N_m} + c_h W_m \\ &= \frac{(1-\rho)c_s}{i_m^{(1)}} + \left\{ \frac{(1-\rho)L_m}{i_m^{(1)}} + \frac{E(S)i_m^{(2)}}{2i_m^{(1)}} + \frac{\lambda\{x^{(1)}E(S^2) + x^{(2)}(E(S))^2\}}{2(1-\rho)} \right\} c_h. \end{aligned} \quad (6.1)$$

Note that the last term is a constant and is independent of the policy adopted. Setting

$$\alpha = (1-\rho)c_s,$$

$$\beta = \frac{\lambda\{x^{(1)}E(S^2) + x^{(2)}(E(S))^2\}}{2(1-\rho)} c_h,$$

and

$$h_m = (1-\rho)c_h L_m + \frac{1}{2}c_h E(S) i_m^{(2)},$$

we can write

$$C_m = c_m + \beta,$$

where the cost function, c_m , is given by

$$c_m = \frac{\alpha + h_m}{i_m^{(1)}}. \quad (6.2)$$

The terms $i_m^{(1)}$, $i_m^{(2)}$ and L_m are all recursive in nature. For example, to obtain $i_{d\ m}^{(1)}$ which is needed to obtain $i_m^{(1)}$, all the terms $i_{d\ k}^{(1)}$, $0 < k < m$ have to

be evaluated beforehand. Also note that once these terms are obtained, evaluating c_m requires very little effort. Hence, in the process of evaluating c_m , we obtain all the information that is needed to quickly calculate c_1 , c_2 ,

..., and c_{m-1} . In a sense, c_m is thus evaluated sequentially, starting with $m=1$.

The recursive nature of the terms $i_m^{(1)}$, $i_m^{(2)}$, ℓ_m and L_m makes it very difficult to characterize c_m with regard to its convexity or unimodality. However, we now show that the function c_m has a special characteristic which enables us to find a global minimum very efficiently. This characteristic is that, in the process of evaluating c_m , $m=1,2,\dots$, the first local minimum encountered is the *global* minimum. This is stated below as Theorem 6.1. Before we state and prove this theorem, we develop some properties that the terms $i_m^{(1)}$, $i_m^{(2)}$, ℓ_m and L_m possess. These properties are not only used to prove Theorem 6.1 but are also used to significantly reduce the amount of computation required to obtain the terms $i_m^{(2)}$, ℓ_m and L_m . The proof of these properties are given in the Appendix.

$$\textbf{Property 1} \quad i_m^{(1)} > i_n^{(1)}, \quad \text{for } m > n, \quad n=1,2,\dots \quad (6.3)$$

$$\textbf{Property 2} \quad h_m > h_n, \quad \text{for } m > n, \quad n=1,2,\dots \quad (6.4)$$

$$\textbf{Property 3} \quad \frac{i_m^{(2)} - i_{m-1}^{(2)}}{i_m^{(1)} - i_{m-1}^{(1)}} = \frac{i_{m-n}^{(2)} - i_{m-n-1}^{(2)}}{i_{m-n}^{(1)} - i_{m-n-1}^{(1)}} + 2(1+i_s^{(1)})n, \quad \text{for } m > n, \quad n=1,2,\dots \quad (6.5)$$

$$\textbf{Property 4} \quad \frac{\ell_m}{i_m^{(1)}} = k, \quad \text{for all } m, \quad (6.6)$$

$$\text{where } k = \frac{1}{\lambda_X^{(1)}}.$$

Property 5
$$\frac{L_m - L_{m-1}}{i_m^{(1)} - i_{m-1}^{(1)}} = \frac{L_{m-n} - L_{m-n-1}}{i_{m-n}^{(1)} - i_{m-n-1}^{(1)}} + (k+E(U))n, \quad \text{for } m>n, \quad n=1,2,\dots$$

(6.7)

where k is as defined in property 4.

Property 6
$$\frac{h_m - h_{n+1}}{i_m^{(1)} - i_{n+1}^{(1)}} > \frac{h_m - h_n}{i_m^{(1)} - i_n^{(1)}}, \quad \text{for } m>n+1, \quad n=1,2,\dots$$

(6.8)

Properties 1,2 and 6 are used to prove Theorem 6.1 and properties 3,4 and 5 are used to calculate $i_m^{(2)}$, l_m and L_m in a more efficient way. Also note that properties 1 and 2 imply that $i_m^{(1)}$ and h_m are both increasing functions with respect to m .

Now we present the main result of this section, which gives a characteristic of the cost function, c_m .

Theorem 6.1 If $c_{k+1} > c_k$, then $c_n > c_k$ for all $n>k>0$.

Proof By contradiction. Suppose there exists an n such that $n>k+1$, and $c_n < c_k$. Then the following inequalities hold:

(i)
$$\frac{\alpha+h_{k+1}}{i_{k+1}^{(1)}} > \frac{\alpha+h_k}{i_k^{(1)}}. \quad (6.9.a)$$

(ii)
$$\frac{\alpha+h_k}{i_k^{(1)}} > \frac{\alpha+h_n}{i_n^{(1)}}. \quad (6.9.b)$$

From (6.3), (6.4) and (6.9.a), we obtain:

$$\frac{h_{k+1}-h_k}{i_{k+1}^{(1)}-i_k^{(1)}} > c_k. \quad (6.10)$$

Similarly, from (6.3), (6.4) and (6.9.b), we obtain:

$$\frac{\alpha+h_n}{i_n^{(1)}} = c_n > \frac{h_{k+1}-h_n}{i_{k+1}^{(1)} i_k^{(1)}} . \quad (6.11)$$

From (6.10), (6.11) and the assumption that $c_n < c_k$, we get:

$$\frac{h_{k+1}-h_k}{i_{k+1}^{(1)} i_k^{(1)}} > \frac{h_n-h_{k+1}}{i_n^{(1)} i_{k+1}^{(1)}} . \quad (6.12)$$

Noting that the numerator and the denominator are positive on both sides of the inequality given in (6.12), we can write:

$$\frac{h_n-h_k}{i_n^{(1)} i_k^{(1)}} > \frac{h_n-h_{k+1}}{i_n^{(1)} i_{k+1}^{(1)}} , \quad (6.13)$$

which contradicts property 6. Thus, $c_n > c_k$ for all $n > k$. ■

Remark 4 In implementation, to calculate $i_m^{(2)}$, l_m and L_m , we use properties 3,4 and 5 instead of using the recursive equations. Note that once $i_m^{(1)}$ is determined, $i_m^{(2)}$, l_m and L_m (and hence, c_m ,) are calculated easily using (6.5), (6.6) and (6.7). Thus, $i_m^{(1)}$ (in fact , $i_{d_m}^{(1)}$) is the only term in which we must use a recursive method for the calculation. This reduces the computational effort significantly.

The above remark, together with Theorem 6.1 produces an extremely simple, and fairly efficient procedure to obtain the optimal policy: sequentially evaluate c_m , starting with $m=1$, until we encounter a point at which the cost function, c_m , begins to rise.

Algorithm to find the optimal m-value

0. Set $k=1$. Compute c_k .
1. Set $k=k+1$. Compute c_k .
2. If $c_k > c_{k-1}$, stop. Optimal value of m is $k-1$.
Otherwise, go to step 1.

7. Numerical Examples

In order to verify the efficiency of our method and to show the superiority of the m-policy over the n-policy, we made extensive numerical tests. Some numerical examples are presented below, comparing the results of the m-policy with those of the n-policy.

It is intuitively obvious that the optimal m-policy is superior to the optimal n-policy since the m-policy uses more information about the state of the queue than the n-policy. In fact, in all the problems we have tested up to date, the m-policy consistently performs better than the n-policy. Due to the complexity of the cost function, however, we could not prove this analytically.

We present four examples below. Example 1 considers the m-policy without vacation and examples 2, 3, and 4 consider the m-policy with vacation. Also while examples 1 and 2 have no start-up times, examples 3 and 4 have start-up times. The vacation times in examples 2 and 3 are assumed to follow a uniform distribution and the vacation time in example 4 is assumed to follow an Erlang distribution.

<Example 1>

$$\begin{aligned}\lambda &= 0.3, E(S)=1, E(S^2)=1.8, \\ x_1 &= 0.25, x_2=0.25, x_3=0.25, x_4=0.25, \\ c_s &= 2000, c_h=3.\end{aligned}$$

<Example 2>

$$\begin{aligned}\lambda &= 0.3, E(S)=1, E(S^2)=1.8, \\ x_1 &= 0.2, x_2=0.3, x_3=0.3, x_4=0.2, \\ c_s &= 1000, c_h=3, \\ V &\text{ follows a uniform distribution on } [5, 10].\end{aligned}$$

<Example 3>

$$\begin{aligned}\lambda &= 0.3, E(S)=1, E(S^2)=1.8, \\ x_1 &= 0.2, x_2=0.3, x_3=0.3, x_4=0.2, \\ E(U) &= 5, E(U^2) = 50, \\ c_s &= 1000, c_h=3, \\ V &\text{ follows a uniform distribution on } [5, 10].\end{aligned}$$

<Example 4>

$$\begin{aligned}\lambda &= 0.2, E(S)=1, E(S^2)=3, \\ x_1 &= 0.3, x_2=0.3, x_3=0.4, \\ E(U) &= 5, E(U^2) = 25, \\ c_s &= 1500, c_h=3, \\ V &\text{ follows a two stage Erlang distribution with mean 2.}\end{aligned}$$

The results of the policy comparisons for these test examples are presented in table 1 through 3. In each table, the values of W_m and C_m are shown for each value of m in the m -policy. The results of the n -policy in the examples have been obtained by the technique which treats a batch as if it were a single unit. As shown in the tables, the optimal m -policy is better

than the optimal n-policy and the minimum points of the cost function C_m occur at the first local minimum points in all three examples. It might be thought that the optimal value of the m-policy should just be the optimal value of the n-policy multiplied by the expected batch size. As we see in tables 2 and 4, this is not necessarily true although in examples 1 and 3 this does happen to be the case. However, in many of the cases that were tested, these two values are observed to be fairly close. Finally, it must be stated that although only two distributions (uniform, Erlang) for the vacation are demonstrated in the examples, many other distributions can be easily implemented.

8. Conclusions

We have introduced two control policies for the $M^x/G/1$ queueing system: the m-policy without vacation which is used for the control of the ordinary $M^x/G/1$ system and the m-policy with vacation which is used for the control of a queueing system where the server takes a sequence of vacations, possibly to do some other work at the end of each busy period. In both of these models, we allowed a start-up interval to represent real systems more accurately. For both policies, we obtained the mean waiting time of an arbitrary unit for a given value of m and developed an efficient procedure to find the stationary optimal m-policy using an important characteristic of the cost function c_m . The direct extension of the m-policy would be the control policy for the system with batch arrivals where one or more queues are served by a single server such as, for example, the model with two queues, studied by Hofri for the case of unit (non-batch) arrivals.

References

- [1] Baba, Y, "On the $M^x/G/1$ Queue with Vacation Time," Opns. Res. Letters, Vol. 5 (1986), pp. 93-98.
- [2] Balachandran, K.R., "Control Policies for a single server system," Mgmt. Sci., Vol. 21 (1975), pp. 1013-1018.
- [3] ____ and Tijms, H.C., "On the D-Policy for the $M/G/1$ Queue," Mgmt. Sci., Vol. 21.(1975), pp. 1073-1076.
- [4] Bell, C. E., "Characterization and Computation of Optimal Policies for Operating an $M/G/1$ Queueing System with Removable Server," Opns. Res., Vol. 19 (1971), pp. 208-218.
- [5] Burke, P. J., "Delays in Single Server Queues with Batch Input," Opns. Res., Vol. 19 (1971), pp. 208-218.
- [6] Cooper, R.B., Introduction to Queueing Theory, Second Ed., Elsevier, North-Holland, New York.
- [7] Fuhrmann, S. W., and Cooper, R. B., "Stochastic Decompositions in the $M/G/1$ Queue with Generalized Vacations," Opns. Res. Vol. 33 (1985), pp. 1117-1129.
- [8] Heyman, D. P., "Optimal Operating Policies for $M/G/1$ Queueing Systems," Opns. Res. Vol. 16 (1968), pp. 362-382.
- [9] ____, "The T-Policy for the $M/G/1$ Queue," Mgmt. Sci., Vol. 23 (1977), pp. 775-778.
- [10] Hofri, M., "Queueing Systems with a Procrastinating Server," Performance 86 and ACM Sigmetrics 1986, Proceedings of the Joint Conference on Computer Performance Modeling, Measurement, and Evaluation, Raleigh, N. Carolina (1986), pp. 245-253.

- [11]Levy, H.,and Kleinrock, L., "A Queue with Starter and a Queue with Vacations: Delay Analysis by Decomposition," Opns. Res. Vol. 34 (1986), pp. 426-436.
- [12]Levy, Y. and Yechiali, U., "Utilization of Idle Time in an M/G/1 Queueing System," Mgmt. Sci. Vol. 22 (1975), pp. 202-211.
- [13]Ramaswami, V., "The N/G/1 Queue and its Detailed Analysis," Adv. Appl. Prob. 12(1980), pp. 222-261.
- [14]Ross, S. M., Stochastic Processes, John Wiley & Sons, 1982.
- [15]Scholl, M. and Kleinrock, L., "On the M/G/1 queue with Rest Period and Certain Service Independent Queueing Disciplines," Opns. Res., Vol. 31 (1983), pp. 705-719.
- [16]Sobel, M.J., "Optimal Average-Cost Policy for a Queue with Start-up and Shut-down Costs," Opns. Res., Vol. 17 (1969), pp. 145-162.
- [17]Yadin, M. and Naor, P., "Queueing Systems with a Removable Service Station," Opns. Res. Quart. Vol. 14 (1963), pp. 393-405.

Table 1.

result of example 1: m-policy without vacation
(without start-up time)

m-policy			n-policy		
m	W_m	C_m	n	W_n	C_n
10	12.09	81.74	1	6.70	220.10
11	12.75	79.94	2	8.37	125.10
12	13.42	78.71	3	10.03	96.77
13	14.08	77.96	4	11.70	85.10
14	14.74	77.57	5	13.37	80.10
*15	15.41	77.48	* 6	15.03	78.43
16	16.07	77.63	7	16.70	78.67
17	16.74	77.99	8	18.37	80.10
18	17.40	78.52	9	20.03	82.32

(* represents the optimal policy)

Table 2.

result of example 2: m-policy with vacation
(without start-up time)

m-policy			n-policy		
m	W_m	C_m	n	W_n	C_n
1	10.43	70.60	1	10.43	70.60
2	10.50	68.78	2	11.14	64.36
3	10.71	66.33	3	12.46	61.45
4	11.09	64.02	* 4	14.01	61.42
5	11.56	62.31	5	15.60	63.03
6	12.04	61.37	6	17.21	65.58
7	12.61	60.82	7	18.83	68.73
* 8	13.21	60.69	8	20.46	72.28
18	13.82	60.89	9	22.09	76.12

(* represents the optimal policy)

Table 3.

result of example 3: m-policy with vacation
(with start-up time)

m-policy			n-policy		
m	W_m	C_m	n	W_n	C_n
1	13.99	66.69	1	13.99	66.69
2	14.08	66.15	* 2	14.68	65.18
3	14.30	65.50	3	15.89	65.36
4	14.65	65.00	4	17.33	67.00
* 5	15.09	64.81	5	18.83	69.53
6	15.53	64.90	6	20.36	72.63
7	16.05	65.26	7	21.92	76.11

(* represents the optimal policy)

Table 4.

result of example 4: m-policy with vacation
(with start-up time)

m-policy			n-policy		
m	W_m	C_m	n	W_n	C_n
8	13.83	117.54	1	7.09	200.67
9	14.95	114.81	2	9.11	152.83
10	16.11	113.00	3	11.36	130.44
11	17.24	111.96	4	13.71	119.31
*12	18.39	111.51	5	16.11	114.09
13	19.55	111.55	*6	18.54	112.36
14	20.71	111.99	7	20.98	112.86

(* represents the optimal policy)

Appendix

Proof of Theorem 4.1:

Theorem 4.1:

$$I_{d,m}(z) = \sum_{j=1}^{m-1} \frac{r_j}{1-r_0} z^j (I_{d,m-j}(z)-1) + \frac{R(z)-r_0}{1-r_0}, \quad m=1,2,\dots$$

Proof By conditioning on the number of units who arrive during the first vacation, we can obtain a recursive equation that yields $P\{I_{d,m}=k\}$ as

$$P\{I_{d,m}=k\} = \sum_{j=0}^{m-1} r_j P\{I_{d,m-j}=k-j\} + r_k, \quad k \geq m, \quad m=1,2,\dots \quad (A.1)$$

If we solve (A.1) in terms of $P\{I_{d,m}=k\}$,

$$P\{I_{d,m}=k\} = \sum_{j=1}^{m-1} r_j' P\{I_{d,m-j}=k-j\} + r_k', \quad k \geq m, \quad m=1,2 \quad (A.2)$$

where $r_j' = \frac{r_j}{1-r_0}$.

From (A.2), the p.g.f. of $I_{d,m}$ is obtained as

$$\begin{aligned} I_{d,m}(z) &= \sum_{k=m}^{\infty} P\{I_{d,m}=k\} z^k \\ &= \sum_{k=m}^{\infty} \left\{ \sum_{j=1}^{m-1} r_j' P\{I_{d,m-j}=k-j\} + r_k' \right\} z^k \\ &= \sum_{j=1}^{m-1} r_j' z^j \sum_{k=m}^{\infty} P\{I_{d,m-j}=k-j\} z^{k-j} + \sum_{k=m}^{\infty} r_k' z^k \\ &= \sum_{j=1}^{m-1} r_j' z^j (I_{d,m-j}(z)-1) + \sum_{k=1}^{\infty} r_k' z^k \\ &= \sum_{j=1}^{m-1} \frac{r_j}{1-r_0} z^j (I_{d,m-j}(z)-1) + \frac{R(z)-r_0}{1-r_0}. \end{aligned}$$

■

Proof of Theorem 5.1

Theorem 5.1:

$$I_{d,m}(z) = \sum_{j=1}^{m-1} x_j z^j \{I_{d,m-j}(z)-1\} + X(z), \quad m=1,2,\dots$$

Proof By conditioning on the number of units in the first arrival, we can obtain a recursive equation that yields $P\{I_{d,m}=k\}$ as

$$P\{I_{d,m}=k\} = \sum_{j=1}^{m-1} x_j P\{I_{d,m-j}=k-j\} + x_k, \quad k \geq m, \quad m=1,2,\dots \quad (A.3)$$

From (A.3), the p.g.f. of $I_{d,m}$ is

$$\begin{aligned} I_{d,m}(z) &= \sum_{k=m}^{\infty} P\{I_{d,m}=k\} z^k \\ &= \sum_{k=m}^{\infty} \sum_{j=1}^{m-1} x_j P\{I_{d,m-j}=k-j\} z^k + \sum_{k=m}^{\infty} x_k z^k \\ &= \sum_{j=1}^{m-1} x_j z^j \sum_{k=m}^{\infty} P\{I_{d,m-j}=k-j\} z^{k-j} + \sum_{k=m}^{\infty} x_k z^k \\ &= \sum_{j=1}^{m-1} x_j z^j \{I_{d,m-j}(z)-1\} + X(z). \end{aligned}$$

■

Proof of Properties 1 through 6

Here we only provide proofs for the m-policy with vacation case. Proofs for the case without vacation can easily be shown analogously. Before proving the properties, we will present some propositions which will be used to prove properties 3, 4 and 5.

Proposition A1

$$\frac{i_{d m}^{(2)} - i_{d m-1}^{(2)}}{i_{d m}^{(1)} - i_{d m-1}^{(1)}} = \frac{i_{d m-n}^{(2)} - i_{d m-n-1}^{(2)}}{i_{d m-n}^{(1)} - i_{d m-n-1}^{(1)}} + 2n, \quad \text{for } m > n, \quad n=1,2,\dots \quad (\text{A.4})$$

proof The proof is by induction on m. If m=2 and n=0, (A.4) is trivially proved. If m=2 and n=1, from (4.8),

$$\frac{i_{d 2}^{(2)} - i_{d 1}^{(2)}}{i_{d 2}^{(1)} - i_{d 1}^{(1)}} = \frac{\frac{1}{1-r_0}(2r_1 i_{d 1}^{(1)} + r_1 i_{d 1}^{(2)}) + \frac{r^{(2)}}{1-r_0} - i_{d 1}^{(2)}}{\frac{1}{1-r_0}(r_1 i_{d 1}^{(1)}) + \frac{r^{(1)}}{1-r_0} - i_{d 1}^{(1)}}$$

(from the fact that $i_{d 1}^{(1)} = \frac{r^{(1)}}{1-r_0}$, $i_{d 1}^{(2)} = \frac{r^{(2)}}{1-r_0}$ and $i_{d 0}^{(1)} = i_{d 0}^{(2)} = 0$)

$$\begin{aligned} & \frac{2i_{d 1}^{(1)} + i_{d 1}^{(2)}}{i_{d 1}^{(1)}} \\ &= \frac{i_{d 1}^{(2)} - i_{d 0}^{(2)}}{i_{d 1}^{(1)} - i_{d 0}^{(1)}} + 2. \end{aligned}$$

So proposition A1 is true if m=2.

For the induction step, suppose (A.4) is true for m=k-1. To verify (A.4) for m=k, we use the following identity, which can be obtained after some algebraic manipulation using (4.8).

$$\frac{i_{d k}^{(2)} - i_{d k-1}^{(2)}}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} - \frac{i_{d k}^{(1)} - i_{d k-1}^{(1)}}{i_{d k-1}^{(1)} - i_{d k-2}^{(1)}} = \frac{1}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} \left[\sum_{j=1}^{k-1} \frac{r_j}{1-r_0} \{ 2j(i_{d k-j}^{(1)} - i_{d k-1-j}^{(1)}) \} \right]$$

$$+ (i_{d k-j}^{(2)} - i_{d k-1-j}^{(2)}) - \frac{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}}{i_{d k-1}^{(1)} - i_{d k-2}^{(1)}} (i_{d k-j}^{(1)} - i_{d k-1-j}^{(1)}) \} \Big]. \quad (A.5)$$

If we apply the induction hypothesis on the right hand side of (A.5), we get

$$\begin{aligned} & \frac{i_{d k}^{(2)} - i_{d k-1}^{(2)}}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} - \frac{i_{d k}^{(1)} - i_{d k-1}^{(1)}}{i_{d k-1}^{(1)} - i_{d k-2}^{(1)}} \\ &= \frac{1}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} \left[\sum_{j=1}^{k-1} \frac{r_j}{1-r_0} \{ 2j(i_{d k-j}^{(1)} - i_{d k-1-j}^{(1)}) + (i_{d k-j}^{(2)} - i_{d k-1-j}^{(2)}) \right. \\ & \quad \left. - \left(\frac{i_{d k-j}^{(2)} - i_{d k-j-1}^{(2)}}{i_{d k-j}^{(1)} - i_{d k-j-1}^{(1)}} + 2(j-1) \right) (i_{d k-j}^{(1)} - i_{d k-1-j}^{(1)}) \} \right] \\ &= \frac{2}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} \sum_{j=1}^{k-1} \frac{r_j}{1-r_0} (i_{d k-j}^{(1)} - i_{d k-1-j}^{(1)}) \\ & \quad \frac{i_{d k}^{(1)} - i_{d k-1}^{(1)}}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} , \quad (A.6) \end{aligned}$$

where the last equality follows from the definition of $i_{d k}^{(1)}$. From (A.6), we obtain

$$\frac{i_{d k}^{(2)} - i_{d k-1}^{(2)}}{i_{d k-1}^{(2)} - i_{d k-2}^{(2)}} - \frac{i_{d k}^{(1)} - i_{d k-1}^{(1)}}{i_{d k-1}^{(1)} - i_{d k-2}^{(1)}} = 2.$$

So under induction hypothesis, (A.6) also holds for $m=k$. Thus, by the principle of mathematical induction, (A.4) holds for any m greater than 0. ■

Proposition A2 $\frac{\ell_{d m}}{i_{d m}^{(1)}} = k, \quad \text{for all } m \geq 1 \quad (\text{A.7})$

where $k = \frac{1}{\lambda_X(1)}$.

proof The proof is by induction on m . If $m=1$, (A.7) is trivially verified by using (4.8.a) and (4.15). For the induction step, suppose (A.7) is true for $m=k-1$. Under induction hypothesis we now prove that (A.7) is also true for $m=k$.

$$\begin{aligned} \frac{\ell_{d k}}{i_{d k}^{(1)}} &= \frac{E(V) + \sum_{j=1}^{k-1} r_j \ell_{d k-j}}{r^{(1)} + \sum_{j=1}^{k-1} r_j i_{d k-j}^{(1)}} \\ &= \frac{k r^{(1)} + k \sum_{j=1}^{k-1} r_j i_{d k-j}^{(1)}}{r^{(1)} + \sum_{j=1}^{k-1} r_j i_{d k-j}^{(1)}} \\ &= k. \end{aligned}$$

So under the induction hypothesis, (A.7) also holds for $m=k$. Thus, by the principle of mathematical induction, we conclude that (A.7) holds for all m greater than 0. ■

Proposition A3

$$\frac{L_{d m} - L_{d m-1}}{i_{d m}^{(1)} - i_{d m-1}^{(1)}} = \frac{L_{d m-n} - L_{d m-n-1}}{i_{d m-n}^{(1)} - i_{d m-n-1}^{(1)}} + kn, \quad \text{for } m > n, \quad n=1,2,\dots \quad (\text{A.8})$$

where k is defined in proposition A2.

proof Proposition A3 can be proved in the same way as proposition A1, using proposition A2 appropriately. ■

We now prove properties 1 through 6. The proofs for Properties 3, 4 and 5 will use the above propositions.

Property 1 $i_m^{(1)} > i_n^{(1)}$, for $m > n$, $n=1,2,\dots$

proof Note that we only need to prove that $i_m^{(1)} > i_{m-1}^{(1)}$ for $m > 0$. This can be easily proved by induction on m and using equation (4.12). ■

Property 2 $h_m > h_n$, for $m > n$, $n=1,2,\dots$

proof By induction, we can prove that $i_m^{(2)}$ and L_m are both increasing functions with respect to m . Since h_m is a positive combination of two increasing functions, namely, $i_m^{(2)}$ and L_m , hence it is also an increasing function. ■

Property 3
$$\frac{i_m^{(2)} - i_{m-1}^{(2)}}{i_m^{(1)} - i_{m-1}^{(1)}} = \frac{i_{m-n}^{(2)} - i_{m-n-1}^{(2)}}{i_{m-n}^{(1)} - i_{m-n-1}^{(1)}} + 2(1+i_s^{(1)})n, \text{ for } m > n, n=1,2,\dots$$

proof Using proposition A1, we have:

$$\begin{aligned} \frac{i_m^{(2)} - i_{m-1}^{(2)}}{i_m^{(1)} - i_{m-1}^{(1)}} &= \frac{i_{d\ m}^{(2)} - i_{d\ m-1}^{(2)} + 2i_s^{(1)}(i_{d\ m}^{(1)} - i_{d\ m-1}^{(1)})}{i_{d\ m}^{(1)} - i_{d\ m-1}^{(1)}} \\ &= \frac{i_{d\ m}^{(2)} - i_{d\ m-1}^{(2)}}{i_{d\ m}^{(1)} - i_{d\ m-1}^{(1)}} + 2i_s^{(1)} \\ &= 2(1+i_s^{(1)}). \end{aligned} \tag{A.9}$$

From (A.9), we obtain property 3 directly. ■

Property 4 $\frac{\ell_m}{i_m^{(1)}} = k$, for all $m \geq 1$,

where $k = \frac{1}{\lambda_X(1)}$.

proof Using proposition A2, we obtain:

$$\begin{aligned}\frac{l_m}{i_m^{(1)}} &= \frac{l_{dm} + E(U)}{i_{dm}^{(1)} + \lambda E(U)x^{(1)}} \\ &= \frac{ki_{dm}^{(1)} + E(U)}{i_{dm}^{(1)} + \lambda E(U)x^{(1)}} \\ &= k.\end{aligned}$$

■

Property 5 $\frac{L_m - L_{m-1}}{i_m^{(1)} - i_{m-1}^{(1)}} = \frac{L_{m-n} - L_{m-n-1}}{i_{m-n}^{(1)} - i_{m-n-1}^{(1)}} + (k + E(U))n$ for $m > n$, $n = 1, 2, \dots$

where k is defined in property 4.

proof Using proposition A3, we have:

$$\begin{aligned}\frac{L_m - L_{m-1}}{i_m^{(1)} - i_{m-1}^{(1)}} &= \frac{L_{dm} - L_{dm-1} + (i_{dm}^{(1)} - i_{dm-1}^{(1)})E(U)}{i_{dm}^{(1)} - i_{dm-1}^{(1)}} \\ &= k + E(U).\end{aligned}\tag{A.10}$$

From (A.10), we obtain property 5 directly.

■

Property 6 $\frac{h_m - h_{n+1}}{i_m^{(1)} - i_{n+1}^{(1)}} > \frac{h_m - h_n}{i_m^{(1)} - i_n^{(1)}}$ for $m > n+1$, $n = 1, 2, \dots$

(A.11)

proof From property 3, we know

$$\begin{aligned}\frac{i_m^{(2)} - i_{m-1}^{(2)}}{i_m^{(1)} - i_{m-1}^{(1)}} &> \frac{i_{m-1}^{(2)} - i_{m-2}^{(2)}}{i_{m-1}^{(1)} - i_{m-2}^{(1)}} > \frac{i_{m-3}^{(2)} - i_{m-2}^{(2)}}{i_{m-2}^{(1)} - i_{m-3}^{(1)}} > \dots\end{aligned}\tag{A.12}$$

To prove property 5, we use the following identity:

If a, b, c, d, e, f , are all positive

$$\frac{a}{b} > \frac{c}{d} > \frac{e}{f} \text{ holds iff } \frac{a}{b} > \frac{a+c}{b+d} > \frac{a+c+e}{b+d+f}, \text{ etc.}$$

If we apply this identity to (A.12), for $m > n+1$, we obtain

$$\frac{i_m^{(2)} - i_{n+1}^{(2)}}{i_m^{(1)} - i_{n+1}^{(1)}} > \frac{i_m^{(2)} - i_n^{(2)}}{i_m^{(1)} - i_n^{(1)}}. \quad (\text{A.13})$$

Similarly, we can prove that

$$\frac{L_m - L_{n+1}}{i_m^{(1)} - i_{n+1}^{(1)}} > \frac{L_m - L_n}{i_m^{(1)} - i_n^{(1)}}. \quad (\text{A.14})$$

Set $h_m = c_1 L_m + c_2 i_m^{(2)}$ where $c_1 = (1-\rho)c_h$ and $c_2 = \frac{1}{2}c_h E(S)$. Since c_1 and c_2 are both positive, from (A.13) and (A.14) we can obtain (A.11). ■