The Optimal Service Policies in an M/G/1 Queue with Consecutive Vacations

Yu Song, *Member, IACSIT*

***Abstract*—In this research, we consider a single server queueing system with Poisson arrivals and multiple vacation types, in which the server can choose one of several types of vacations to take when he finishes serving all customers in the system. Upon completion of a vacation, the server may either take another vacation with a certain probability or check the number of customers waiting in the system. In the latter case, if the number of customers is greater than a critical threshold, the server will resume serving the queue exhaustively; otherwise, he will take another vacation. A variety of vacation types are available and the choice is the discretion of the server. The cost structure consists of a constant waiting cost rate, fixed costs for starting up service, and reward rates for taking vacations. It is shown that this infinite buffer queueing system can be formulated as a finite state Semi-Markov decision process. With this finite state model, we can determine the optimal service policy to minimize the long-term average cost of this vacation system. Some practical stochastic production and inventory control systems can be effectively studied using this model.**

***Index Terms*—Queueing systems, vacation models, threshold policies, semi-Markov decision process.**

I. Introduction

Queueing systems with server's vacations (also called vacation models) have been studied widely due to their applications in a variety of production and telecommunication systems. In such a system, the server will become unavailable for a random period of time when the customers are waiting in the systen1 at a service completion instant. This random period of time, often called a vacation, may represent the time period that the server will perform a job of another type. Vacation models can be used to study the waiting line system where the server can process multiple types of jobs.

There are various vacation models due to different policies about starting a vacation and starting the service of the queue. Doshi[1], [2] provides surveys on vacation models. Takagi [3] also presents the analyses of a variety of queueing systems with vacations and further analyses are available from the references therein. One subset of vacation models are the single server vacation models with a threshold service resumption policy. In such a system, the server leaves for a vacation period of random length whenever the system becomes empty at a service completion instant. When the server returns from the vacation, he will check the number of customers waiting in the syste1n. If the number of waiting custo1ners at, this instant is greater than a threshold value, the server will start serving the customers until the system is empty; otherwise, the server will go for another vacation (Note that a vacation can equivalently be regarded as another type of job which the server can do). Vacation models with the threshold policies have attracted many researchers' attention over the past decade. Kella [4] studied this type of service policy in an M/G/ 1 queue with vacations of a single type and introduced a si1nple algorithm for determining the optional threshold value of this system with a linear waiting cost function. Federgruen and so [5] showed the optimality of a single threshold policy for the same system with more general waiting cost function.

Vacation models with multiple vacation types were first studied by Zhang, Vickson and Eegine [6]. They investigated an M/G/1 queue with two vacation types and a two threshold policy and developed a finite search algorithm to find the optimal values of the two thresholds, although the optimality of this type of policy was not proven. Zhang and Love [7] studied the M/G/1 queue with exceptional first vacation and threshold policy and derived a bound for searching the global optimal threshold value to minimize the long-run average operating cost. Zhang, Vickson, and Love [8] investigated the M/G/1 queue with multiple vacation types and a service resumption policy or the threshold type using Semi-Markov Decision Process which is a generalization of the vacation models studied in [6] and [7]. Zhang, Love and Song [9] expanded the model to systems with stochastically available vacations.

However, all previous vacation models assumed that the server must go back to check the queue after a vacation. This assumption may limit the applicability of this class of vacation models. In most real queueing system with vacations, the vacation usually represents some kind of secondary jobs that are performed by the server when the primary jobs (the queue) are all finished (i.e. empty) at a point in time. These secondary jobs can be classed into different types. Moreover, since there usually exists a set-up cost to resume normal queue service, it could be reasonable to do another secondary job with without checking the queue after a vacation.

In this study, we consider an M/G/1 queueing system multiple vacation types, in which the server can choose one of several types of vacations to take when he finishes serving all customers in the system. Upon completion of a vacation, the server may either take another vacation with a certain probability or check the number of customers waiting in the system. Here we formulate a semi-Markov decision process for this vacation model in order to find an optimal server's policy, which minimizes the long-term average cost of the system. The policy should specify (i) when the server resumes serving the queue (i.e., the threshold value M); and (ii) for each state in which the queue length is less than M, what type

Manuscript received December 10, 2013; revised January 16, 2014.

Y. Song is with Business School, Manchester Metropolitan University, on leave from Department of System Management, Faculty of Information Engineering, Fukuoka Institute of Technology, Fukuoka, Japan (e-mail: song@fit.ac.jp).

of vacation the server should try to take.

The remainder of this paper is organized as follows. Section II describes the queueing system with multi-vacation types and Section III formulates a semi-Markov decision process (SMDP) for the system. Because of the nature of the threshold policy, we can obtain a finite state space SMDP for this vacation model with an infinite buffer. Section IV presents an algorithm to find the optimal threshold value M for serving the queue and the optimal schedule for performing other N types of jobs. In Section V we present some numerical examples of this vacation model. These examples serve to demonstrate that such multiple threshold policies provide selection rules as to the optimal vacation types to choose as well as the criteria for service resumption. In the final section, we conclude this study with some summary comments.

II. Model Description

In this study, we consider the following M/G/1 queueing system.

Customers arrive at the system according to a Poisson process with an arrival rate of λ.

The service time S is a random variable with general distribution. The average service time is $\overline{S}$.

The server serves the queue exhaustively when it attends the queue.

When the system becomes empty, the server can select a vacation of type n, whose random time is denoted by V_n where n = 1, 2, 3, ..., N. We use "$\geq_{st}$" to stand for "stochastically greater than or equal to". It is assumed that $V_1 \geq_{st} V_2 \geq_{st} V_3 \geq_{st} \ldots \geq_{st} V_N$.

Upon completion of the first vacation, the server may either take a second vacation of type m with probability q_{nm} or check the number of customers waiting in the system with probability q_{n0}

where $q_{n0} = 1 - \sum_{m=1}^{N} q_{nm}$.

Upon completion of the second vacation, the server must go back to check the number of customers waiting in the system.

If the number of waiting customers equals or exceeds a specified number (M) at the instant that the server checks the queue, the queue is served immediately. If the number lies between 0 and M - 1, the server is then free to select a vacation of type n' he or she wishes to take, as the policy defined in 4) above.

The reward of performing a type n vacation job is r_n, where it is assumed $r_n \geq r_{n+1}$ for any n.

The start-up cost for the server to resume normal service is denoted by r_0.

The holding cost per unit time of a customer in the queue is h.

III. The SMDP Model

We formulate the M/G/1 queue with N vacation types and a threshold M policy as a semi-Markov decision process as follows:

A. The State Space

Since the decision instants are the vacation completion instants or the service completion instant, when the system is empty, we can use one variable to describe the state of the system.

X = {0', 0, 1, 2, 3, ... , M - 1}. X = 0' represents that the system is empty at a service completion instant and $X = i$, where i = 0, 1, 2, ... , M - 1, represents that the system has i customers at a vacation completion instant.

B. The Action Set

For the state X = 0' or 0, the action set is A{0'} = A(0) = {1, 2, 3, ... , N}. An action $a = n$ means that the server takes a type n vacation For the state $X = i$, where i = 1, 2, 3, ... , M - 1, the action set is A(i) = {0, 1, 2, ... , M}, where a= 0 represents that the server starts serving the customers exhaustively.

C. The Transition Probabilities of the Process

At any state $i \in X$, the server can take a type n vacation, where n = 1, 2, 3, ... , N. The next state (at the next decision epoch) i, will be either 0' or the non-empty state j where j <= M - 1, depending on the number of customers arriving during this vacation. There are three cases as follows:

Case 1: Transition $i \rightarrow 0'$ (a service completion state) given $a = n > 0$ at state i.

This is the case when the number of customers arriving during the vacation is more than $M-1-i$. Based on assumption 4) in the last section, the server will serve all customers immediately and exhaustively after the vacation. So 0' state will be reached. The probability of this case is:

$$p_{i0'}(a = n) = 1 - \sum_{k=0}^{M-1-i} u_{n,k}, \tag{1}$$

where

$$u_{n,k} = v_{n,0} + \sum_{m=1}^{N} q_{nm} v_{m,k} + v_{n,1} + \sum_{m=1}^{N} q_{nm} v_{m,k-1} + \cdots + v_{n,k} + \sum_{m=1}^{N} q_{nm} v_{m,0} \tag{2}$$

$$v_{n,j} = \int_0^{\infty} e^{-\lambda t} \frac{(\lambda t)^j}{j!} dFV_n(t) \tag{3}$$

And is the probability that the number of arrivals during a type n vacation is j.

Case 2: Transition $i \rightarrow j$ (a vacation completion state) given $a = n > 0$ at state i.

This is the case when the number of customers arriving during the vacation is between 0 and $M-1-i$.

$$p_{ij}(a=n) = v_{n,j-i} \tag{4}$$

where $i \leq j \leq M-1$.

Case 3: Transition i → 0' given a= 0 at state i.

$$p_{i0'}(a=0) = 1 \tag{5}$$

This is the case when the server resumes service at state i.

D. The Expected Transition Times

Based on the conditional probability argument, we obtain

the expected transition times as follows:

$$\begin{aligned}\tau_i(a=n) &= \overline{U}_n + \sum_{k=M-i}^{\infty} k u_{n,k}\bar{\theta} + i\bar{\theta}\sum_{k=M-i}^{\infty} u_{n,k} \\ &= \overline{U}_n + \lambda\overline{U}_n\bar{\theta} - \sum_{k=0}^{M-i-1} k u_{n,k}\bar{\theta} + i\bar{\theta}(1-\sum_{k=M-i}^{\infty} u_{n,k}) \\ &= \frac{\overline{U}_n}{1-\rho} - \sum_{k=0}^{M-i-1} k u_{n,k}\bar{\theta} + i\bar{\theta}(1-\sum_{k=M-i}^{\infty} u_{n,k})\end{aligned} \tag{6}$$

where

$$\begin{aligned}\overline{U}_n &= q_n\overline{V}_n + (1-q_n)q_{n+1}\overline{V}_{n+1} + \cdots \\ &+ \prod_{j=n}^{N-2}(1-p_j)q_{N-1}\overline{V}_{N-1} \\ &+ (1-q_n - \sum_{m=n}^{N-2}\prod_{n}^{m}(1-q_j)q_{m+1})V_N\end{aligned}$$

and

$$\tau_i(a=0) = i\bar{\theta} \tag{7}$$

with $\bar{\theta} = \bar{S}/(1-\rho)$.

E. The One-Step Expected Costs

The cost structure imposed on the system includes a linear holding cost, *h*, for customers in the queue, a reward rate r_n for performing a type *n* vacation (optional job), and a start-up cost r_0 for the server to resume service. Note that r_0 may include a fixed cost incurred whenever the server is shut down. Conditioning on the length of a vacation period of type *n* and the number of arrivals during this period and using the property of Poisson arrival process, we can obtain the one-step expected cost of taking an action of $a = n$ as follows

$$\begin{aligned}C_i(a=n) &= \frac{\lambda h}{2(1-\rho)}U_n^{(2)} + (\lambda\alpha + \frac{hi}{1-\rho})\overline{U}_n - r_n\overline{U}_n \\ &+ \frac{h\theta}{2}(i^2 - \sum_{k=0}^{M-i-1} u_{n,k}(k+i)^2 + \alpha(i - \sum_{k=0}^{M-i-1} u_{n,k}(k+i)\end{aligned}$$

And the one-step expected cost of resuming service when there are *i* customers waiting in the system is

$$C_i(a=0) = \frac{h\bar{\theta}}{2}(i^2 - i) + iC_1^1 + r_0 \tag{8}$$

For the details of the derivation of the above formulas, see Zhang and Love [7]. With the formulae above, the specification for the SMDP is complete.

IV. The Procedure for Finding the Optimal Policy

Since the SMDP has a discrete finite state space and a discrete finite action set, there exists a constant *g* and a non-negative function $u(i), i \in X$ which satisfies the optimality equation

$$\begin{aligned}u(i) = \min_a\{C_i(a) - g(R)\tau_i(a) + \\ \sum_{j=i}^{M-1} p_{ij}(a)u(j) + p_{i0'}(a)u(0')\}\end{aligned} \tag{9}$$

Thus we can find the optimal stationary service policy for the server using a policy-improvement algorithm for a given threshold value *M.* To determine both the optimal *M* and the optimal service policy, the following procedure is suggested.

Algorithm for Finding the Optimal M and Service Policy for the Vacation Model

Step 1: Choose a reasonable value *M*.

Step 2: Choose a stationary policy *R*.

Step 3: For the current rule *R*., compute average costs g(R) and the relative values $u(i), i \in X$, as the unique solution to the linear equations

$$\begin{aligned}u(i) = \min_a\{C_i(a) - g(R)\tau_i(a) + \\ \sum_{j=i}^{M-1} p_{ij}(a)u(j) + p_{i0'}(a)u(0')\}, i \in X\end{aligned}$$

and

$$u(s) = 0$$

where *s* is an arbitrarily chosen state.

Step 4: For each state $i \in X$, determine an action a_i yielding the minimum in

$$\begin{aligned}\min_{a\in A(i)}\{C_i(a) - g(R)\tau_i(a) + \\ \sum_{j=i}^{M-1} p_{ij}(a)u(j) + p_{i0'}(a)u(0')\}.\end{aligned}$$

The new stationary policy R' is obtained by choosing $R'_i = a_i$ for all $i \in X$ with the convention that R'_i is chosen equal to the old action R'_i when this action minimizes the policy-improvement quantity.

Step 5: If the new policy *R'* equals the old policy, go to Step 6. Otherwise, go to Step 3 with R replaced by *R*'.

Step 6: If a *M*-1 = 0, find the minimum state *i* <=M -1 in which ai= 0, the algorithm is stopped with optimal policy R and optimal threshold value *M* = *i.* Otherwise, let *M* = *M* + 1 and *Ri* = *R'i*, *i* =0', 0, 1, ..., *M*- 1, and RM = 0, go to Step 3.

Note that the initial value of *M* and the initial stationary policy in the first two steps of this procedure can be chosen as the optimal two threshold policy with two of N vacation types using the method developed by Zhang *et al*., [6]. Numerical tests indicate using this stationary policy as the Starting position can reduce the computational time of finding the optimal policy.

TABLE I: The Basic Parameters

E(*S*)	λ	E(V_1)	E(V_2)	E(V_3)
1	0.6	3	2	1
h	r_0	r_1	r_3	r_3
2	20	13	12	11

(All random variables are exponential)

TABLE II: Probabilities to Take a Second Vacation

	q_{n0}	q_{n1}	q_{n2}	q_{n3}
Case 1	1	0	0	0
Case 2	0.1	0.3	0.3	0.3
Case 3	0.1	0.5	0.3	0.1
Case 4	0.1	0.1	0.3	0.5

V. Numerical Examples

Below in Table I we provide a basic data set for a case of three types of vacation jobs. The interarrival time of customers to the system, the servicing of these customers as well as the service times for the three vacation job types are assumed to be exponentially distributed.

For numerical tests, we set 4 cases with different probabilities to take a second vacation (Table II). For example, in Case 1, $q_{n1}=q_{n2}=q_{n3}=0$ means the server cannot take a second vacation. The server must go back to check the queue with probability $q_{n1}(=1)$. In case 4, $q_{n1}=0.1$ implies that the probabilities to take a type 1 vacation as its second vacation is 0.1.

In Table III, we find optimal policies (based on the algorithm of the preceding section) with $M = 5$.

TABLE III: OPTIMAL POLICY FOR M=5

State (i)	0'	0	1	2	3	4
Vacation to take ($A(i)$)						
Case 1	1	1	1	1	1	1
Case 2	3	3	1	1	1	1
Case 3	1	1	1	1	1	1
Case 4	3	3	1	1	1	2

As explained above, Case 1 is a special case because there is no chance to take a second vacation. This is same as the model discussed in [8]. Table III shows that the decision is to take the stochastically largest vacation. In case 2, where the probabilities to take a second vacation are the same for all vacations, the server may select a smaller vacation when the queue is much shorter than the threshold value M. This results from that the server improves his return by taking a second vacation when the queue is short enough. This can also be observed in Case 4. In this case, the probability to take a stochastically small vacation is quite large.

Analysis of these tables and other numerical results leads us to make the following observations:

Without consecutive vacations, the optimal policy has the multi-threshold policy structure wherein the smaller the queue of waiting customers, the larger (stochastically) is vacation type to be taken. While we cannot prove this, the observation supports the conjecture that the multi-threshold policy is the optimal for such vacation models with multiple vacation types. This optimal policy pattern has also been discovered in the case where the vacations are stochastically available (see Zhang, Love and Song [9]).

On the contrary, when consecutive vacations are possible, the smallest vacation type may be taken when the queue is short. The overall return could be improved by taking the second vacation.

In certain cases, some vacation types may not be included in the optimal policy (in particular when M is small). For example in Case 1, vacation types 2 and 3 are not present. This indicates that, in the interests of minimizing the long-run average cost, types 2 and 3 jobs should not be performed unless the reward rate for performing such jobs is increased. The proposed sensitivity procedure can be used to compute this minimum reward rate.

If the reward rates are the same for all vacation types and there is no inspection cost after completion of each vacation, the server will always take the stochastically smallest vacation type (i.e., the smallest job).

The service resumption threshold M yields a convex total cost curve, typical of such queuing phenomenon.

VI. SUMMARY

In this research we have formulated an SMDP structure to represent an M/G/1 queueing system with multiple vacation types. With this we are able to construct an algorithm to determine both a service redemption policy for the server as well as the optimal rules regarding choice of vacation types to be utilized.

It is easy to see that this model could be used to study various practical problems. We can also study the problem of how to achieve the appropriate server's utilization level given an average arrival rate, the service rate, and the set of preventive maintenance jobs and their availability probability vector.

In this study, we assume that the server can only take two consecutive vacations at most. However, it is trivial to generalize the model to any finite numbers of consecutive vacations.

REFERENCES

[1] B. T. Doshi, "Queueing systems with vacations: A survey," *Queueing Systems* 1, pp. 29-66, 1986.

[2] B. T. Doshi, "Single-server queues with vacations," in: H. Takagi (Ed.) *Stochastic Analysis of Computer and Communication Systems*, North-Holland, Amsterdam, pp. 217-265, 1990.

[3] H. Takagi, *Queueing Analysis - A Foundation of Performance Evaluation*, Elsevier, Amsterdam, vol. 1, 1991.

[4] O. Kella, "The threshold policy in the M/G/1 queue with server vacations," *Naval Research Logistics*, vol. 36, pp.111-123, 1989.

[5] A. Federgruen and K. C. So, "Optimality of threshold policies in single server queueing system with vacations," *Advances in Applied Probability*, vol. 23, pp. 388-405, 1991.

[6] Z. G. Zhang, R. G. Vickson, and M. J. A. van Eenige, "Optimal two threshold policies in an M/G/1 queue with two vacation types," *Performance Evaluation*, vol. 29, pp. 63-80, 1997.

[7] Z. G. Zhang and C. E. Love, "The threshold policy in an M/G/l queue with an exceptional first vacation," *INFOR*. vol. 36, no. 4, pp. 193-204, 1998.

[8] Z. G. Zhang, R. Vickson, and C. E. Love, "The optimal service policies in an M/G/1 queueing system with multiple vacation types," *INFOR*, vol. 39, pp. 357-366, 2001.

[9] Z. G. Zhang, C. E. Love, and Y. Song, "The optimal service time allocation of a versatile serer to queue jobs and stochastically available non-queue jobs of different type," *Computer and Operations Research*, vol. 34, pp. 1857-1870, 2007.

Yu Song was born in China. He earned his Ph.D. at Tohoku University, Japan in 1992. Currently he is a professor of Fukuoka Institute of Technology, Japan. He is a senior member of IACSIT.

He majors in operational research, especially queueing theory and heuristics for combinatorial optimization.