JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **8**, Number **1**, February **2012**

doi:10.3934/jimo.2012.8.1

pp. **1–17**

A HEURISTIC ALGORITHM FOR THE OPTIMIZATION OF M/M/s QUEUE WITH MULTIPLE WORKING VACATIONS

Chia-Huang Wu

Department of Industrial Engineering and Management
National Chiao Tung University
Hsingchu 30010, Taiwan

Kuo-Hsiung Wang

Department of Business Administration
Asia University
Wufeng, Taichung 41354, Taiwan

Jau-Chuan Ke

Department of Applied Statistics
National Taichung University of Science and Technology
Taichung 404, Taiwan

Jyh-Bin Ke

Department of Applied Mathematics
National Chung-Hsing University
Taichung 402, Taiwan

(Communicated by Wuyi Yue)

Abstract. This paper focuses on an M/M/s queue with multiple working vacations such that the server works with different service rates rather than no service during the vacation period. We show that this is a generalization of an M/M/1 queue with working vacations in the literature. Service times during vacation period, or during service period and vacation times are all exponentially distributed. We obtain the useful formula for the rate matrix **R** through matrix-geometric method. A cost function is formulated to determine the optimal number of servers subject to the stability conditions. We apply the direct search algorithm and Newton-Quasi algorithm to heuristically find an approximate solution to the constrained optimization problem. Numerical results are provided to illustrate the effectiveness of the computational algorithm.

1. Introduction.

We analyze an M/M/s queue with multiple working vacations such that the server works with variable service rates rather than completely terminates service during a vacation period. Such a vacation is called a *working vacation* (WV) [14]. The server starts a working vacation when the system is empty. When a working vacation ends and there are no customers in the system, the server begins another working vacation. If the server returns from a working vacation and find the not-empty system, he switches to another service rate. The time interval between two successive vacations is called a service period [14, p. 107].

2000 *Mathematics Subject Classification.* Primary: 60K10, 60K25; Secondary: 90B22, 90B25.
Key words and phrases. Newton-Quasi algorithm; optimization; rate matrix; sensitivity analysis; working vacations.

It is assumed that customers arrive according to a Poisson process with rate λ. The service times during a service period follow exponential distribution with mean $1/\mu_B$. The service times during a vacation period follow another exponential distribution with mean $1/\mu_v$. When there are no waiting customers in the system, the server begins a working vacation of mean duration $1/\eta$, where vacation times are also exponentially distributed. We assume that arriving customers form a single waiting line based on the order of their arrivals; that is, the first-come, first-served discipline is followed. Suppose that one server can serve only one customer at a time. Customers entering into the service facility and finding that the server is busy have to wait in the queue until the server is available. We also assume that the customer is always assigned or shifted to the server with higher service rate if it is available.

Servi and Finn [14] introduced an Internet Protocol (IP) access network example where each gateway router which connected to the optical network could be formulated as an M/M/1/WV queue. The example illustrated that the use of wavelength division multiplexing (WDM) access network can improve performance in reconfiguration of the wavelengths. A reconfigurable WDM optical access network as Figure 1 is considered. Each access routers with multiple ports are connected to the global IP network through a gateway router. Each port can transfer data over a single wavelength. The wavelength reconfiguration problem is to determine which access router ports should be connected to the gateway router ports according to its wavelengths. A strategy is to reconfigure the additional roving wavelengths to the next router when one access router has empty queue. It is assumed that there are s roving wavelengths (s servers) with the capacity of mean service rate u^*. For keeping the connection, each access router serves permanently at a nominal average rate $s\mu_v$ and all additional roving wavelength are initially configured to access router 1 with a total average service rate of $s\mu_v + su^* = s\mu_B$. At a service completion, one additional roving wavelength would be assigned to the next router if no more requirement (customer) waiting in the queue. This cycle continually repeats itself.

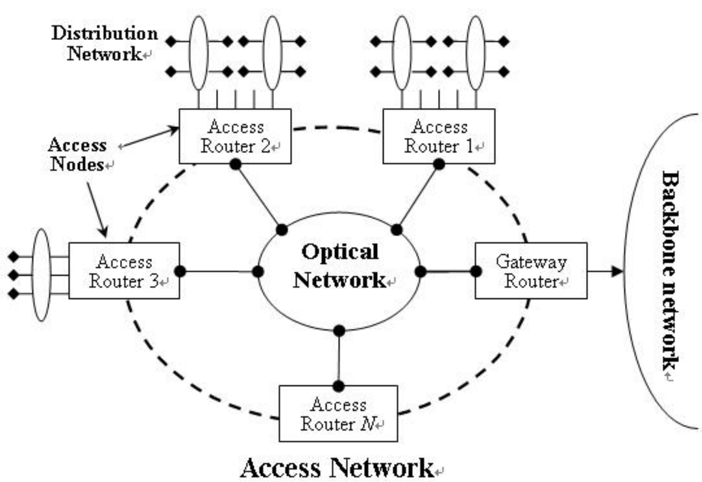

FIGURE 1. An optical access Network

For example, there are $s = 5$ roving wavelengths which is tunable in the access network. Initially, access router 1 and 2 serve with service rates $5\mu_B$ and $5\mu_v$, respectively. When the number of customers in router 1 changes from 5 to 4 (one idle sever), one roving wavelength would be assigned to router 2. This results in access router 1 instantaneously dropping its average service rate from $5\mu_B$ to $5\mu_v + 4u^* = \mu_v + 4\mu_B$ and access router 2 is operating at an average service rate of $5\mu_v + u^* = 4\mu_v + \mu_B$. We consider an optical network with s WDM to configure different roving wavelengths (s servers) which can be allocated to several access routers. From the viewpoint of access router i, it follows an M/M/s queue with working vacation policy. Next, the investigation of queueing systems with server vacations or working vacations are reviewed.

During the last two decades, the queueing systems with server vacations or multiple working vacations have been investigated by many researchers. Past work may be divided into two categories: (i) the category of server vacations, and (ii) the category of multiple working vacations. In the first category the readers may refer to the survey paper by Doshi [6]. The GI/M/1 queue with server vacations have been studied by several authors such as Karaesmen and Gupta [8], Chatterjee and Mukherjee [4], Tian et al. [16], etc. They considered that the vacation time follows exponential distribution in [8], [16] and general distribution in [4], respectively. Karaesmen and Gupta [8] analyzed a finite capacity GI/M/1 queue with server vacations. They developed the queue length distribution at arrival and random epochs for the multiple vacations case. The model was previously investigated by Chatterjee and Mukherjee [4] for infinite capacity queue considering server vacations. They applied the embedded Markov chain technique to obtain the steady-state probability distributions of system size at pre-arrival and at random epochs, separately. Fuhrmann [7] studied a single server queue of M/G/1 type with server vacations. The finite capacity M/G/1 queue with server vacations was investigated by Lee [9]. In the second category Servi and Finn [14] first examined an M/M/1 queue with working vacations, where inter-arrival times, service times during service period, service times during vacation period, and vacation times are all exponentially distributed. The queueing model is denoted by M/M/1/WV. Recently, Liu et al. [12] used stochastic decomposition method to analyze M/M/1/WV queue which is different from Servi and Finn's method. Discrete time GI/Geo/1 queue with multiple working vacations was first studied by Li et al. [10]. Later Wu and Takagi [18] extended Servi and Finn's M/M/1/WV queue to an M/G/1/WV queue. They assumed that service times during service period, service times during vacation period as well as vacation times are all generally distributed. Baba [1] extended Servi and Finn's M/M/1/WV queue to a GI/M/1/WV queue and derived the steady-state system length distributions at arrival and arbitrary epochs. Banik et al. [2] studied a finite capacity GI/M/1 queue with multiple working vacations. They developed some important system performance measures such as, the probability of blocking and the expected waiting time in the system, etc. Recently, Lin and Ke [11] studied a Markovian queueing system with a single working vacation. They provided a closed-form method for obtaining the rate matrix. In the study of working vacations, existing literature focuses mainly on a single server queue with multiple vacations. This would motivate us to investigate the M/M/s/WV queue.

In Section 2, the steady-state equations are solved via matrix-geometric method. The computation for rate matrix $\mathbf{R}$ follows in Section 3. This queueing model is a generalization of an M/M/1 queue with working vacations shown in Section 4.

In Section 5, the explicit formulae for system performance measures are developed. Cost analysis and sensitivity investigation are presented in Section 6. Finally, Section 7 summarizes the paper with some concluding remarks.

2. **Steady-state results.** The states of the system are described by the pair (k, n), $k = 0, 1, 2, \ldots, s-1, s$ and $n = k, k+1, k+2, \ldots$, where k represents the number of working servers and n represents the number of customers in the system. In steady-state, we introduce the following notations:

$P_0(n) \equiv$ probability that there are n customers in the system when all servers are on a working vacation, where $n = 0, 1, 2, \ldots$;
$P_1(n) \equiv$ probability that there are n customers in the system when there are $(s-1)$ servers on a working vacation, where $n = 1, 2, \ldots$;
$P_k(n) \equiv$ probability that there are n customers in the system when there are $(s-k)$ servers on a working vacation, where $n = k, k+1, k+2, \ldots$, $k = 2, 3, 4, \ldots, s$.

The steady-state equations for $P_k(n)(k = 0, 1, 2, \ldots, s)$ relating to Figure 2 are given by:
(i) $k = 0$

$$\lambda P_0(0) = \mu_B P_1(1) + \mu_v P_0(1), \tag{1}$$
$$(\lambda + n\mu_v + s\eta)P_0(n) = \lambda P_0(n-1) + (n+1)\mu_v P_0(n+1), \quad 1 \le n \le s-1 \tag{2}$$
$$(\lambda + s\mu_v + s\eta)P_0(n) = \lambda P_0(n-1) + s\mu_v P_0(n+1), \quad n \ge s \tag{3}$$

(ii) $1 \le k \le s-1$

$$(\lambda + k\mu_B)P_k(k) = (k+1)\mu_B P_{k+1}(k+1) + (k\mu_B + \mu_v)P_k(k+1) + (s-k+1)\eta P_{k-1}(k), \tag{4}$$

$$[\lambda + k\mu_B + (n-k)\mu_v + (s-k)\eta]P_k(n) = [k\mu_B + (n-k+1)\mu_v]P_k(n+1) + \lambda P_k(n-1) + (s-k+1)\eta P_{k-1}(n), \quad k+1 \le n \le s-1 \tag{5}$$

$$[\lambda + k\mu_B + (s-k)\mu_v + (s-k)\eta]P_k(n) = [k\mu_B + (s-k)\mu_v]P_k(n+1) + \lambda P_k(n-1) + (s-k+1)\eta P_{k-1}(n), \quad n \ge s \tag{6}$$

(iii) $k = s$

$$(\lambda + s\mu_B)P_s(s) = s\mu_B P_s(s+1) + \eta P_{s-1}(s), \tag{7}$$
$$(\lambda + s\mu_B)P_s(n) = \lambda P_s(n-1) + s\mu_B P_s(n+1) + \eta P_{s-1}(n), \quad n \ge s+1. \tag{8}$$

In the next section, we apply the matrix-geometric method to solve (1) to (8) for $P_k(n)$, where $k = 0, 1, 2, \ldots, s$ and $n = k, k+1, k+2, \ldots$. A matrix geometric method was introduced by Neuts [13] for analyzing many complex queueing systems. Using this method, we derive the steady-state probabilities $P_k(n)(k = 0, 1, 2, \ldots, s)$.

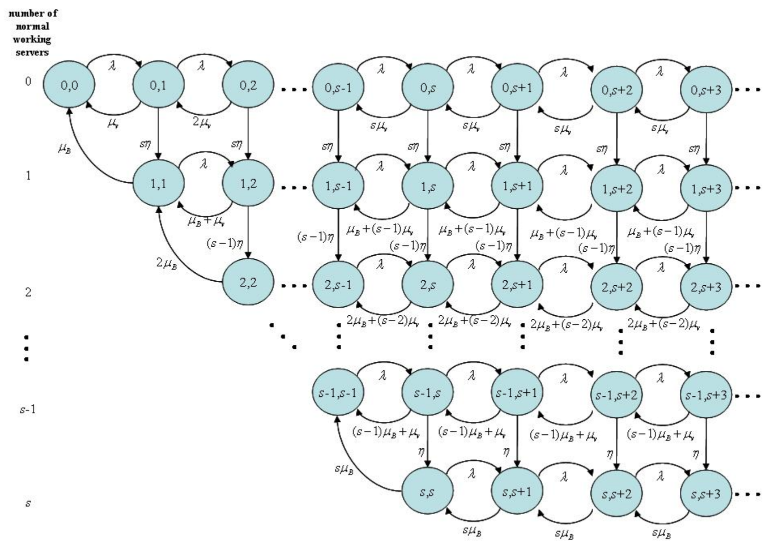

FIGURE 2. State-transition-rate diagram for an M/M/s/WV queue

The infinitesimal generator for the quasi birth and death process has the block-tridiagonal form (see Neuts [13]):

$$\mathbf{Q} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{C} & & & & & & & \\ \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{C} & & & & & & \\ & \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C} & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & \mathbf{A}_{s-1} & \mathbf{B}_{s-1} & \mathbf{C} & & & \\ & & & & \mathbf{A}_s & \mathbf{B}_s & \mathbf{C} & & \\ & & & & & \mathbf{A}_s & \mathbf{B}_s & \mathbf{C} & \\ & & & & & & \mathbf{A}_s & \mathbf{B}_s & \mathbf{C} \\ & & & & & & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{9}$$

Each entry of the matrix $\mathbf{Q}$ is a square matrix of order $s+1$ listed as follows:

$$\mathbf{C} = -\lambda \mathbf{I}, \tag{10}$$

$$\mathbf{B}_0 = \begin{pmatrix} \lambda & & & & & \\ -\mu_B & \lambda+\mu_B & & & & \\ & -2\mu_B & \lambda+2\mu_B & & & \\ & & \ddots & \ddots & & \\ & & & -(s-1)\mu_B & \lambda+(s-1)\mu_B & \\ & & & & -s\mu_B & \lambda+s\mu_B \end{pmatrix}, \tag{11}$$

$$\mathbf{B}_i = \begin{pmatrix} b_{i,0} & & & & & \\ & b_{i,1} & & & & \\ & & b_{i,2} & & & \\ & & & \ddots & & \\ & & & & b_{i,s-1} & \\ & & & & & b_{i,s} \end{pmatrix}, \quad i = 1, 2, \ldots, s \tag{12}$$

$$\mathbf{A}_i = \begin{pmatrix} a_{i,0} & -s\eta & & & & \\ & a_{i,1} & -(s-1)\eta & & & \\ & & a_{i,2} & -(s-2)\eta & & \\ & & & \ddots & \ddots & \\ & & & & a_{i,s-1} & -\eta \\ & & & & & a_{i,s} \end{pmatrix}, i = 1, 2, \ldots, s \tag{13}$$

where $\mathbf{I}$ is the identity matrix of order $s+1$, and

$$a_{i,j} = \begin{cases} -i\mu_v + j\mu_B, & 1 \le i+j \le s \\ -[(s-j)\mu_v + j\mu_B], & i+j > s \end{cases} \tag{14}$$

$$b_{i,j} = \begin{cases} \lambda + i\mu_v + j\mu_B + (s-j)\eta, & 1 \le i+j \le s \\ \lambda + (s-j)\mu_v + j\mu_B + (s-j)\eta, & i+j > s. \end{cases} \tag{15}$$

Let $\mathbf{P}$ denote the corresponding steady-state probability vector of $\mathbf{Q}$. By partitioning the vector $\mathbf{P}$ as $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{s-1}, \mathbf{P}_s, \mathbf{P}_{s+1}, \ldots]$, where $\mathbf{P}_n = [P_0(n), P_1(n+1), P_2(n+2), \ldots, P_s(n+s)]$ is a row vector with dimension $(s+1)$, we can write the steady-state equations $\mathbf{PQ} = \mathbf{0}$ and normalization condition as

$$\mathbf{P}_0\mathbf{B}_0 + \mathbf{P}_1\mathbf{A}_1 = \mathbf{0} \tag{16}$$

$$\mathbf{P}_{n-1}\mathbf{C} + \mathbf{P}_n\mathbf{B}_n + \mathbf{P}_{n+1}\mathbf{A}_{n+1} = \mathbf{0}, \quad n = 1, 2, \ldots, s-1 \tag{17}$$

$$\mathbf{P}_{n-1}\mathbf{C} + \mathbf{P}_n\mathbf{B}_s + \mathbf{P}_{n+1}\mathbf{A}_s = \mathbf{0}, \quad n = s, s+1, s+2, \ldots \tag{18}$$

$$\sum_{n=1}^{s-1}\mathbf{P}_n\mathbf{e} + \mathbf{P}_s(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e} = \mathbf{1} \tag{19}$$

where $\mathbf{R}$ is the minimal non-negative solution to the matrix quadratic equation (20) and $\mathbf{e}$ is a column vector with dimension $(s+1)$ whose transpose is $[1, 1, \ldots, 1]$.

$$\mathbf{R}^2\mathbf{A}_s + \mathbf{R}\mathbf{B}_s + \mathbf{C} = \mathbf{0}. \tag{20}$$

Using (16)-(17) and (19), the steady-state vectors $[\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_s]$ can be solved by the block Gauss-Seidel iteration method. The rest steady-state vector $[\mathbf{P}_{s+1}, \mathbf{P}_{s+2}, \mathbf{P}_{s+3}, \ldots]$ can be determined recursively by $\mathbf{P}_n = \mathbf{P}_s\mathbf{R}^{n-s}$, for $n \ge s$. Once the rate matrix $\mathbf{R}$ is determined, the steady-state solutions $[\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{s-1}, \mathbf{P}_s, \mathbf{P}_{s+1}, \ldots]$ are obtained.

3. **Computation for rate matrix R.** To obtain $[\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{s-1}, \mathbf{P}_s, \mathbf{P}_{s+1}, \ldots]$, it is necessary to solve the minimal solution of (20). Based on the upper triangular structures of matrices $\mathbf{A}_s$, $\mathbf{B}_s$, and $\mathbf{C}$, the rate matrix solution $\mathbf{R}$ is also upper triangular. For the multiple-server queue, we assume that there exists a matrix $\mathbf{R}$ satisfying the following equation:

$$\mathbf{R}^2\mathbf{A}_s + \mathbf{R}\mathbf{B}_s + \mathbf{C} = \mathbf{0},$$

where $\mathbf{A}_s$, $\mathbf{B}_s$, and $\mathbf{C}$ are given in (13), (12), (10), respectively.

Using the Maple computer program and matrix algorithm, we develop the explicit formula for rate matrix $\mathbf{R}$ as follows:

$$\mathbf{R} = \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \cdots & \cdots & t_{1,s+1} \\ 0 & t_{2,2} & t_{2,3} & \cdots & \cdots & t_{2,s+1} \\ & & \ddots & & \vdots & \vdots \\ & & & \ddots & \vdots & \vdots \\ & & & & t_{s,s} & t_{s,s+1} \\ & & & & 0 & t_{s+1,s+1} \end{pmatrix}, \tag{21}$$

where

$$\begin{aligned} t_{i,i} &= \frac{\lambda + \theta_i + (s-i+1)\eta - \sqrt{[\lambda + \theta_i + (s-i+1)\eta]^2 - 4\lambda\theta_i}}{2\theta_i}, \text{for } 1 \le i \le s+1 \\ t_{i,j} &= \frac{\sum_{m=i+1}^{j-1} t_{i,m}t_{m,j}\theta_j + (s-j+2)\eta \sum_{m=i}^{j-1} t_{i,m}t_{m,j-1}}{\lambda + \theta_j + (s-j+1)\eta - (t_{i,i} + t_{j,j})\theta_j}, \text{for } 1 \le i \le j \le s+1 \\ t_{i,j} &= 0, \text{ for } i > j, \text{and} \\ \theta_j &= (s-j+1)\mu_v + (j-1)\mu_B. \end{aligned}$$

Note that $t_{i,j}$ is the corresponding eigenvalue of the rate matrix $\mathbf{R}$. Once the rate matrix $\mathbf{R}$ is determined, the steady-state solutions can be evaluated.

4. **Generalization of Servi and Finn's result.** The queueing model considered in this paper is more general than the work of Servi and Finn [14] who first introduced an M/M/1/WV queue. In this section, we will show that our queueing model generalizes the M/M/1/WV queue. For a singer server queue (that is, $s = 1$), we obtain from (20) that

$$\mathbf{R}^2\mathbf{A}_1 + \mathbf{R}\mathbf{B}_1 + \mathbf{C} = \mathbf{0}, \tag{22}$$

It yields from (10), (12)-(13) that

$$\mathbf{A}_1 = \begin{pmatrix} -\mu_v & -\eta \\ 0 & -\mu_B \end{pmatrix}, \mathbf{B}_1 = \begin{pmatrix} \lambda + \mu_v + \eta & 0 \\ 0 & \lambda + \mu_B \end{pmatrix}, \mathbf{C} = \begin{pmatrix} -\lambda & 0 \\ 0 & -\lambda \end{pmatrix}.$$

Since $\mathbf{A}_1$, $\mathbf{B}_1$, $\mathbf{C}$ are 2×2 upper triangle matrices. Therefore we set $\mathbf{R} = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$. From (22), we get

$$\begin{pmatrix} a & b \\ 0 & d \end{pmatrix}\begin{pmatrix} a & b \\ 0 & d \end{pmatrix}\begin{pmatrix} -\mu_v & -\eta \\ 0 & -\mu_B \end{pmatrix} + \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}\begin{pmatrix} \lambda + \mu_v + \eta & 0 \\ 0 & \lambda + \mu_B \end{pmatrix} + \begin{pmatrix} -\lambda & 0 \\ 0 & -\lambda \end{pmatrix} = 0. \tag{23}$$

Solving (23) yields

$$a = \frac{\lambda + \mu_v + \eta - \sqrt{(\lambda + \mu_v + \eta)^2 - 4\lambda\mu_v}}{2\mu_v}, b = \frac{\eta a^2}{\mu_B(1-a)}, d = \frac{\lambda}{\mu_B}.$$

Thus, a rate matrix $\mathbf{R}$ is obtained. Let $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1]$ and $\mathbf{P}_0 = [P_0(0), P_1(1)]$. Since $\mathbf{PQ} = \mathbf{0}$ and $\mathbf{P}_1 = \mathbf{P}_0\mathbf{R}$, it implies from (16) that

$$\mathbf{P}_0\mathbf{B}_0 + \mathbf{P}_1\mathbf{A}_1 = \mathbf{P}_0\mathbf{B}_0 + \mathbf{P}_0\mathbf{R}\mathbf{A}_1 = \mathbf{P}_0(\mathbf{B}_0 + \mathbf{R}\mathbf{A}_1) = \mathbf{0}. \tag{24}$$

We can get $\mathbf{P}_0 = [P_0(0), P_1(1)]$ using (24) and the normalization condition of

$$\mathbf{P}_0(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = \mathbf{1}, \tag{25}$$

where $\mathbf{I}$ is the identity matrix of order 2, and $\mathbf{e}$ is a column vector whose transpose is $[1,1]$. We find that

$$\mathbf{B}_0 = \begin{pmatrix} \lambda & 0 \\ -\mu_B & \lambda+\mu_B \end{pmatrix}, (\mathbf{I}-\mathbf{R})^{-1} = \frac{1}{(1-a)(1-d)} \begin{pmatrix} 1-d & -b \\ 0 & 1-a \end{pmatrix}.$$

Solving (24) and (25), we obtain

$$\begin{aligned} (\lambda-\mu_v a)P_0(0) - \mu_B P_1(1) &= 0, && (26) \\ (-\eta a - b\mu_B)P_0(0) + \mu_B P_1(1) &= 0, && (27) \\ (1+b-d)P_0(0) + (1-a)P_1(1) &= (1-a)(1-d). && (28) \end{aligned}$$

Solving (26)-(28) again, after the algebraic manipulation, we get

$$P_0(0) = \frac{(1-a)(\mu_B-\lambda)}{\mu_B - a\mu_v}. \tag{29}$$

where $a = \frac{\lambda+\mu_v+\eta-\sqrt{(\lambda+\mu_v+\eta)^2-4\lambda\mu_v}}{2\mu_v} = \frac{\lambda z^*}{\mu_v}$. Note that z^* is the smaller root of the equation

$$A(z) = \lambda z^2 - (\lambda+\eta+\mu_v)z + \mu_v = \lambda(z-z^*)(z-\hat{z}) = 0, \text{ for } |\hat{z}| \geq 1 \tag{30}$$

which is (A.4) of [14, p. 49]. Following Servi and Finn's result, we have $\lambda z^* \hat{z} = \mu_v$. It implies that $a = \lambda z^*/\mu_v = 1/\hat{z}$. Substituting $a = 1/\hat{z}$ into (29) yields

$$P_0(0) = \frac{(\mu_B-\lambda)(1-1/\hat{z})}{\mu_B - \mu_v/\hat{z}} = \frac{(\mu_B-\lambda)(\hat{z}-1)}{\mu_B\hat{z} - \mu_v}. \tag{31}$$

Thus from Servi and Finn's equation (1.2) (see [14, p. 42]), we obtain

$$\begin{aligned} Pr(N=0) &= r(1-\lambda/\mu_B) + (1-r)(1-\hat{z}^{-1}) \\ &= \frac{\hat{z}-1}{\hat{z}} - \frac{(\lambda\hat{z}-\mu_B)(\hat{z}-1)(\lambda\hat{z}-\mu_v)\mu_B}{\mu_B\hat{z}(\mu_B\hat{z}-\mu_v)(\lambda\hat{z}-\mu_B)} \\ &= \frac{\hat{z}-1}{\hat{z}}\left(1 - \frac{\lambda\hat{z}-\mu_v}{\mu_B\hat{z}-\mu_v}\right) = \frac{(\hat{z}-1)(\mu_B-\lambda)}{\mu_B\hat{z}-\mu_v} \end{aligned}$$

which is in agreement with expression (31).

5. **System performance measures and some numerical examples.**

The system performance measures, such as the average number of customers in the system (L_s), the average number of servers during service period ($E[NW]$), and the average number of servers during vacation period ($E[WV]$), can be obtained from the steady-state probabilities $\mathbf{P}_n = [P_0(n), P_1(n+1), P_2(n+2), \ldots, P_s(n+s)]$ as follows:

$$\begin{aligned} L_s &= \sum_{n=0}^{\infty} \mathbf{P}_n \begin{pmatrix} n \\ n+1 \\ n+2 \\ \vdots \\ n+s \end{pmatrix} = \sum_{n=0}^{\infty} \mathbf{P}_n(\mathbf{v}+n\mathbf{e}) \\ &= \mathbf{P}_0\mathbf{v} + \mathbf{P}_1(\mathbf{v}+\mathbf{e}) + \cdots + \mathbf{P}_{s-1}[\mathbf{v}+(s-1)\mathbf{e}] + \mathbf{P}_s(\mathbf{v}+s\mathbf{e}) \\ &\quad +\mathbf{P}_s\mathbf{R}[\mathbf{v}+(s+1)\mathbf{e}] + \cdots \\ &= \sum_{n=0}^{s-1} \mathbf{P}_n(\mathbf{v}+n\mathbf{e}) + \mathbf{P}_s(\mathbf{v}+s\mathbf{e}) + \mathbf{P}_s\mathbf{R}[\mathbf{v}+(s+1)\mathbf{e}] \\ &\quad +\mathbf{P}_s\mathbf{R}^2[\mathbf{v}+(s+2)\mathbf{e}] + \cdots \\ &= \sum_{n=0}^{s-1} \mathbf{P}_n(\mathbf{v}+n\mathbf{e}) + \mathbf{P}_s(\mathbf{I}+\mathbf{R}+\mathbf{R}^2+\cdots)(\mathbf{v}+s\mathbf{e}) \\ &\quad +\mathbf{P}_s(\mathbf{R}+2\mathbf{R}^2+3\mathbf{R}^3+\cdots)\mathbf{e} \\ &= \sum_{n=0}^{s-1} \mathbf{P}_n(\mathbf{v}+n\mathbf{e}) + \mathbf{P}_s(\mathbf{I}-\mathbf{R})^{-1}(\mathbf{v}+s\mathbf{e}) + \mathbf{P}_s\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e}, \end{aligned} \tag{32}$$

$$\begin{aligned} E[NW] &= \sum_{n=0}^{\infty} \mathbf{P}_n \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ s \end{pmatrix} = \sum_{n=0}^{\infty} \mathbf{P}_n\mathbf{v} = \sum_{n=0}^{s-1} \mathbf{P}_n\mathbf{v} + \mathbf{P}_s(\mathbf{I}+\mathbf{R}+\mathbf{R}^2+\cdots)\mathbf{v} \\ &= \sum_{n=0}^{s-1} \mathbf{P}_n\mathbf{v} + \mathbf{P}_s(\mathbf{I}-\mathbf{R})^{-1}\mathbf{v}, \end{aligned} \tag{33}$$

$$E[WV] = s - E[NW] \tag{34}$$

where $\mathbf{v}$ and $\mathbf{e}$ are column vectors with dimension $(s+1)$ whose transposes are denoted by $[0, 1, 2, \ldots, s]$ and $[1, 1, \ldots, 1]$, respectively. Moreover, the expected customers served by normal working servers ($E[L_sNW]$), the expected customers served by working vacation servers ($E[L_sWV]$), and the expected idle working vacation servers that do not serve customers currently ($E[IDWV]$) are also obtained as follows:

$$E[L_sNW] = \sum_{n=0}^{\infty} \mathbf{P}_n \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ s \end{pmatrix} = \sum_{n=0}^{s-1} \mathbf{P}_n\mathbf{v} + \mathbf{P}_s(\mathbf{I}-\mathbf{R})^{-1}\mathbf{v} = E[NW], \tag{35}$$

$$
\begin{aligned}
& E[L_sWV] \\
= \;& \mathbf{P}_1 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 0 \end{pmatrix} + \mathbf{P}_2 \begin{pmatrix} 2 \\ 2 \\ \vdots \\ 2 \\ 1 \\ 0 \end{pmatrix} + \cdots + \mathbf{P}_s \begin{pmatrix} s \\ s-1 \\ \vdots \\ 2 \\ 1 \\ 0 \end{pmatrix} + \mathbf{P}_{s+1} \begin{pmatrix} s \\ s-1 \\ \vdots \\ 2 \\ 1 \\ 0 \end{pmatrix} + \cdots \\
= \;& \sum_{n=1}^{s-1} \mathbf{P}_n \begin{pmatrix} \max\{\min\{n,s\},0\} \\ \max\{\min\{n,s-1\},0\} \\ \vdots \\ \max\{\min\{n,s-(s-2)\},0\} \\ \max\{\min\{n,s-(s-1)\},0\} \\ \max\{\min\{n,s-s\},0\} \end{pmatrix} + \mathbf{P}_s(\mathbf{I}-\mathbf{R})^{-1} \begin{pmatrix} s \\ s-1 \\ \vdots \\ 2 \\ 1 \\ 0 \end{pmatrix}, \quad (36)
\end{aligned}
$$

$$
\begin{aligned}
E[IDWV] &= \mathbf{P}_0 \begin{pmatrix} s \\ s-1 \\ \vdots \\ 2 \\ 1 \\ 0 \end{pmatrix} + \mathbf{P}_2 \begin{pmatrix} s-1 \\ s-2 \\ \vdots \\ 1 \\ 0 \\ 0 \end{pmatrix} + \cdots + \mathbf{P}_{s-1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&= \sum_{n=0}^{s-1} \mathbf{P}_n \begin{pmatrix} \max\{0,s-n\} \\ \max\{0,s-n-1\} \\ \vdots \\ \max\{0,s-n-(s-2)\} \\ \max\{0,s-n-(s-1)\} \\ \max\{0,s-n-s\} \end{pmatrix}. \quad (37)
\end{aligned}
$$

It could be verify that $E[L_sNW] + E[L_sWV] + E[IDWV] = s$. (Therefore, $E[L_sWV] + E[IDWV] = E[WV]$).

Consider an Internet Protocol (IP) access network with s roving wavelengths, some numerical results about the number of customers in the access router (L_s) are presented graphically on the basis of the following three cases:

Case 1. $\mu_B = 3.5$, $\mu_v = 2.0$, $\eta = 0.05$, vary the values of λ from 1.0 to 3.0.
Case 2. $\lambda = 1.5$, $\mu_B = 3.5$, $\eta = 0.05$, vary the values of μ_v from 0.5 to 3.0.
Case 3. $\lambda = 1.5$, $\mu_v = 2.0$, $\eta = 0.05$, vary the values of μ_B from 2.5 to 4.5.

Results of L_s are depicted in Figures 3-5 for cases 1-3, respectively. Figure 3 shows that L_s drastically increases as λ increases for $s = 1$, while L_s slightly increases as λ increases for $s \geq 2$. Figure 4 illustrates that L_s drastically decreases as μ_v increases for $s = 1$, while L_s slightly decreases as μ_v increases for $s \geq 2$. From Figure 5, we find that L_s slightly decreases as μ_B increases for any s. It is noted that the more roving wavelengths, the more flexible the optical access network is. Also, diving and transferring the additional wavelength partly would be more elastic than shifting all the additional wavelength at a time.

6. **Cost analysis and sensitivity investigation.** A steady-state expected cost function per unit time is developed. Suppose the average arrival rate is λ, Our

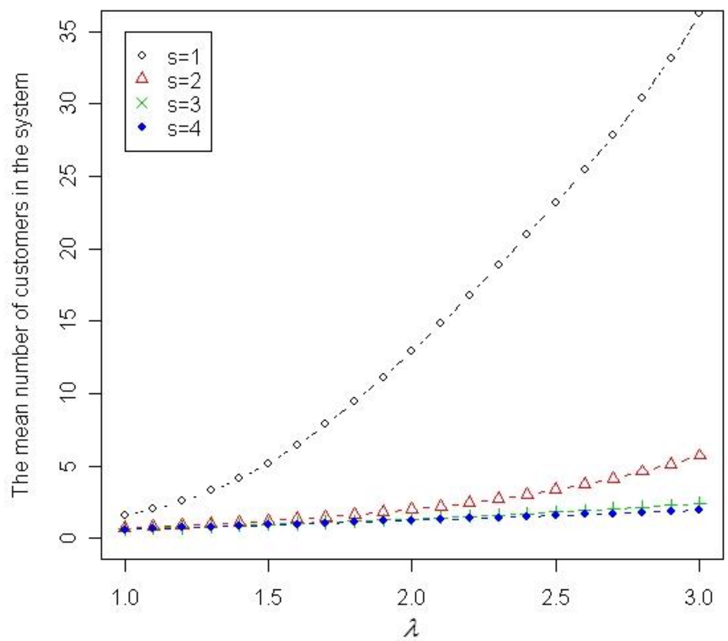

FIGURE 3. The mean number of customers in the system versus λ

object is to determine the value of the number of tunable wavelengths s, nominal average service rate μ_v, and fully average service rate μ_B under stability condition $\rho = \lambda/s\mu_B < 1$ and a natural (rational) constraint $\mu_B = \mu_v + u^* \geq \mu_v$. The discrete variable s is required to be a positive integer, and the continuous variables μ_v and μ_B are positive numbers. The following cost parameters are considered:

C_h	$\equiv$	holding cost per unit time per customer present in the system;
C_B	$\equiv$	cost per unit time when one server is normal working;
C_V	$\equiv$	cost per unit time per customer served by one working vacation server;
C_I	$\equiv$	cost per unit when one server is idle on working vacation state;
C_s	$\equiv$	cost per unit time of providing a specific server rate.

The cost minimization problem can be written as

$$\min_{s,\mu_v,\mu_B} F(s, \mu_v, \mu_B) \tag{38}$$

subject to

$$s\mu_B > \lambda \text{ and } \mu_B \geq \mu_v, \tag{39}$$

where

$$F(s, \mu_v, \mu_B) = C_h L_s + C_B E[L_s NW] + C_V E[L_s WV] + C_I E[IDWV] + C_s(\mu_B + \mu_v). \tag{40}$$

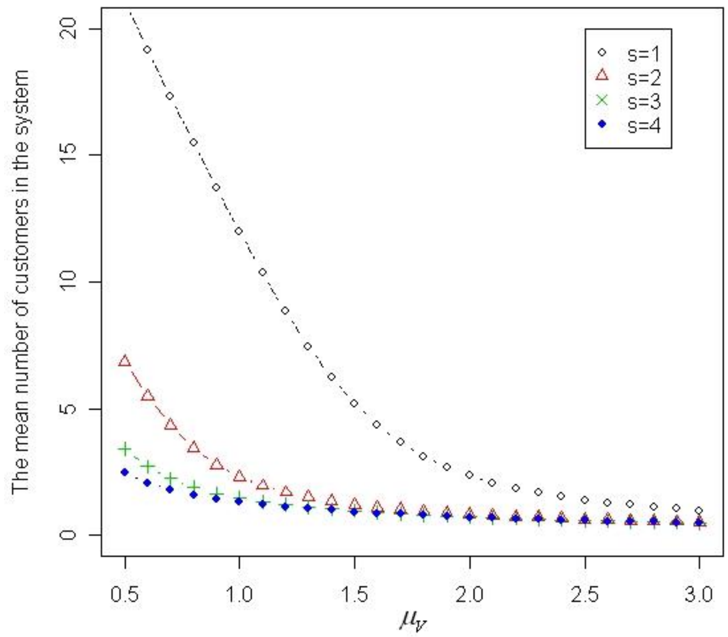

FIGURE 4. The mean number of customers in the system versus μ_v

(see White et al. [17], p. 218). It is extremely difficult to prove the convex property and then develop the analytical results for the optimum values (s^*, μ_v^*, μ_B^*) because the cost function is highly complex.

Due to the fact s is a discrete quantity, μ_v and μ_B are continuous quantities, we attempt to obtain an approximate (local) minimum solution $(\hat{s}, \hat{\mu}_v, \hat{\mu}_B)$ of (s^*, μ_v^*, μ_B^*) by Newton-Quasi method for various values of s. Newton's method is one of the most powerful and well-known numerical methods for solving the root-finding problems. Consequently, Newton's method is employed to solve the root of $G(\mathbf{x}) = \nabla F(s, \mu_v, \mu_B) = 0$ which is the first order necessary condition of local minimum solution of function G. From Burden and Douglas [3], the sequence

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \nabla G(\mathbf{x}_n)^{-1} G(\mathbf{x}_n) = \mathbf{x}_n - [H(\mathbf{x}_n)]^{-1} \nabla F(\mathbf{x}) = 0 \tag{41}$$

could help us to obtain an approximate minimum solution where $H(\mathbf{x}_n)$ denotes the Hessian matrix of function F. Therefore, equation (41) could be regard as the Newton's method that aims at the root-finding problem of the first derivation equation. Based on Chong and Zak [5], the steps of Newton-Quasi algorithm can be described as follows for any iteration

Step 1. For each value of s, let $\boldsymbol{\mu}_n = [\mu_v, \mu_B]^T$.
Step 2. Set the initial trial solution for $\boldsymbol{\mu}_n$ with $n = 0$.

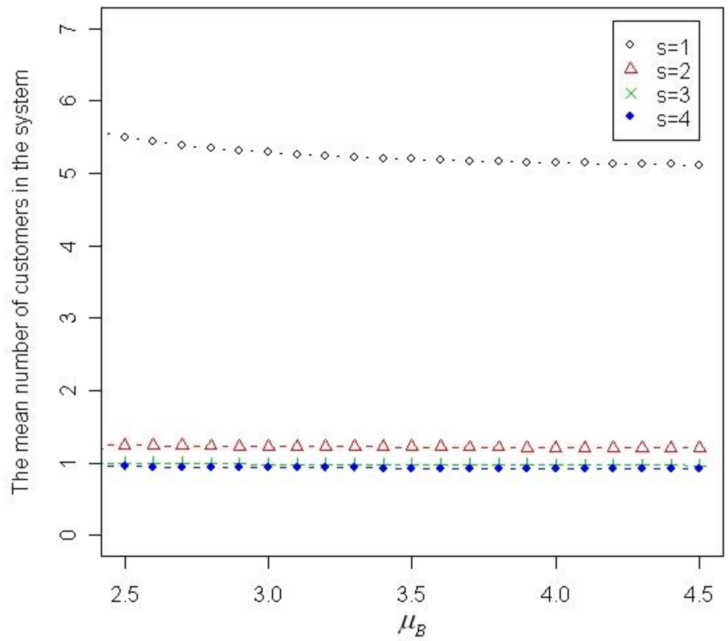

FIGURE 5. The mean number of customers in the system versus μ_B

Step 3. Compute $F(s, \boldsymbol{\mu}_n)$, the cost gradient $\nabla F(s, \boldsymbol{\mu}_n) = [\partial F/\partial \mu_v, \partial F/\partial \mu_B]^T|_{\boldsymbol{\mu}_n}$ and the cost Hessian matrix

$$H(s, \boldsymbol{\mu}_n) = \left(\begin{array}{cc} \partial^2 F/\partial \mu_v^2 & \partial^2 F/\partial \mu_v \partial \mu_B \\ \partial^2 F/\partial \mu_B \partial \mu_v & \partial^2 F/\partial \mu_B^2 \end{array} \right)\Bigg|_{\boldsymbol{\mu}_n}.$$

Step 4. Find the new trial solution $\boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_n - [H(s, \boldsymbol{\mu}_n)]^{-1} \cdot \nabla F(s, \boldsymbol{\mu}_n)$.

Step 5. Set $n = n + 1$ and repeat Steps 3-4 if $|\partial F/\partial \mu_v| > \varepsilon_1$ or $|\partial F/\partial \mu_B| > \varepsilon_2$ or $\|\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n\| > \varepsilon_3$, where ε_1, ε_2, and ε_3 are the tolerances; otherwise, go to Step 6.

Step 6. Find the local minimum value $F(s, \boldsymbol{\mu}_n) = F(s, \mu_v^*, \mu_B^*)$.

Because the cost function is complex, the gradient ∇F and the Hessian matrix H may be derived (approximated) numerically while their expressions could not be explicitly obtained. The gradient vector could be computed directly as

$$\nabla F(s, \mu_v, \mu_B) \approx \left[\frac{F(s, \mu_v + \Delta, \mu_B) - F(s, \mu_v, \mu_B)}{\Delta}, \frac{F(s, \mu_v, \mu_B + \Delta) - F(s, \mu_v, \mu_B)}{\Delta} \right]^T \tag{42}$$

with Δ is small (For example, 10^{-6}). The elements of Hessian matrix are approximated as

$$\frac{\partial^2 F(s,\mu_v,\mu_B)}{\partial \mu_v^2} \approx \frac{\partial F(s,\mu_v+\varepsilon,\mu_B)/\partial\mu_v - \partial F(s,\mu_v,\mu_B)/\partial\mu_v}{\varepsilon}$$
$$\approx \frac{\frac{F(s,\mu_v+\varepsilon+\Delta,\mu_B)-F(s,\mu_v+\varepsilon,\mu_B)}{\Delta} - \frac{F(s,\mu_v+\Delta,\mu_B)-F(s,\mu_v,\mu_B)}{\Delta}}{\varepsilon} \tag{43}$$

$$\frac{\partial^2 F(s,\mu_v,\mu_B)}{\partial \mu_v \partial\mu_B} \approx \frac{\partial F(s,\mu_v+\varepsilon,\mu_B)/\partial\mu_B - \partial F(s,\mu_v,\mu_B)/\partial\mu_B}{\varepsilon}$$
$$\approx \frac{\frac{F(s,\mu_v+\varepsilon,\mu_B+\Delta)-F(s,\mu_v+\varepsilon,\mu_B)}{\Delta} - \frac{F(s,\mu_v,\mu_B+\Delta)-F(s,\mu_v,\mu_B)}{\Delta}}{\varepsilon} \tag{44}$$

with ε is small. For simplification, the equations (43) and (44) become

$$\frac{\partial^2 F(s,\mu_v,\mu_B)}{\partial \mu_v^2} \approx \frac{F(s,\mu_v+2\Delta,\mu_B) - 2F(s,\mu_v+\Delta,\mu_B) + F(s,\mu_v,\mu_B)}{\Delta^2} \tag{45}$$

$$\frac{\partial^2 F(s,\mu_v,\mu_B)}{\partial \mu_v \partial\mu_B} \approx \frac{F(s,\mu_v+\Delta,\mu_B+\Delta) - F(s,\mu_v+\Delta,\mu_B)}{\Delta^2}$$
$$- \frac{F(s,\mu_v,\mu_B+\Delta) - F(s,\mu_v,\mu_B)}{\Delta^2} \tag{46}$$

while we set $\varepsilon = \Delta$. Similarly, $\partial^2 F/\partial\mu_B\partial\mu_v$ and $\partial^2 F/\partial\mu_B^2$ could be derived easily by the exchange of symbols μ_B and μ_v.

Note that Newton-Quasi method aims at unconstrained optimization problem while our problem has nonnegative constraint and two functional constraints. First, the constraint $s\mu_B^* > \lambda > 0$ often holds because the queueing system would boil up and cause the rapid increase of L_s (F) if the stability condition is violated. Some adjustments steps are provided to handle the following three situations:

Case 1. $\mu_B^* > 0$, $\mu_v^* < 0$, set $\mu_v^* = 0$, re-apply Newton's method to find μ_B^*.
Case 2. $\mu_B^* \geq \mu_v^* \geq 0$, the optimization solution is feasible.
Case 3. $\mu_v^* > \mu_B^* > \lambda/s$, set $\mu_v = \mu_B = t$, re-apply Newton's method to find t^*.

For Cases 1 and 3, we simplify the original problem to the approximate optimization problem with single variable, (i.e., discuss the optimal solution at binding equation).

As the example of IP access network mentioned in Section 1, for an access router with mean arrival rate $\lambda = 5$ customers/minute, mean working vacation time $1/\eta =$ 2 minutes, and the following cost parameters (per minute):

$$C_h = \$30/\text{customer},\ C_B = \$180/\text{customer},\ C_V = \$45/\text{customer},$$

$$C_I = \$15/\text{server},\ C_s = \$30/\text{unit}.$$

The Newton-Quasi method is employed to deal with this wavelength reconfiguration problem and some numerical illustrations are provided. Table 1 presents the approximated optimal solutions and the corresponding system performance measures under various numbers of roving wavelengths (servers). From Table 1, it is observed that the minimum cost is 566.347 if the IP access network implements exhaustive schedule (assigns all tunable wavelengths to the next router when the queue becomes empty). The minimum expected cost and the optimal solution are 416.863 and $(\hat{s}, \hat{\mu}_v, \hat{\mu}_B) = (3, 3.70716, 3.70716)$ respectively. That is, for this access router, providing three tunable wavelengths is appropriate for cost saving.

TABLE 1. The approximate value $(\hat{\mu}_v^*, \hat{\mu}_B^*)$ under various values of servers $(\lambda = 5.0, \eta = 0.5)$

$\hat{s}$	1	2	3	4	5	6
$\hat{\mu}_v^*$	6.30991	4.39560	3.70716	3.36074	2.95602	2.61867
$\hat{\mu}_B^*$	7.27773	4.39560	3.70716	3.68261	3.96625	4.23769
F	566.347	427.706	**416.863**	433.770	454.700	474.614
L_s	2.75094	1.68139	1.50040	1.47136	1.49100	1.50565
$E[L_sNW]$	0.28704	0.36595	0.47377	0.55777	0.61370	0.65668
$E[WV]$	0.71296	1.63405	2.52623	3.44223	4.38630	5.34332
$E[L_sWV]$	0.46136	0.77155	0.87497	0.87658	0.86803	0.84668
$E[IDWV]$	0.25160	0.86250	1.65126	2.56565	3.51827	4.49664

The result of Newton-Quasi algorithm in searching the optimal solution is shown in Table 2. In Table 2 (I) (the first stage), the optimal solution found by using Newton-Quasi method is Case 3 ($\mu_v^* > \mu_B^* > \lambda/s$). Therefore, some adjustment procedures (set $\mu_v = \mu_B = t$) are executed as shown in Table 2 (II). It appears from Table 2 that the Newton-Quasi method is working well and converges very fast.

TABLE 2. Newton-Quasi algorithm in searching the optimal solution ($s = 3$, $\lambda = 5.0$, $\eta = 0.5$, $\boldsymbol{\mu} = [5.0/3 + 1.0, 5.0/3 + 2.0]^T$)

(I) Newton-Quasi algorithm Stage I

Iteration	1	2	3	4	5	6
$\hat{\mu}_v^*$	2.66666	3.55229	3.88561	3.92150	3.92185	3.92185
$\hat{\mu}_B^*$	3.66666	3.61194	3.55529	3.54783	3.54776	3.54776
F	427.458	417.317	416.598	416.591	416.591	416.591
$\partial F/\partial \hat{\mu}_v$	-20.5784	-4.36434	-0.38011	-0.00367	-3×10^{-7}	-8×10^{-14}
$\partial F/\partial \hat{\mu}_B$	-9.85432	-1.71118	-0.12899	-0.00125	-1×10^{-7}	-2×10^{-14}
L_s	1.98542	1.57542	1.47034	1.46023	1.46013	1.46013
$E[L_sNW]$	0.61936	0.50486	0.47733	0.47479	0.47477	0.47477
$E[WV]$	2.38064	2.49514	2.52267	2.52521	2.52523	2.52523
$E[L_sWV]$	1.02338	0.89421	0.85005	0.84547	0.84543	0.84543
$E[IDWV]$	1.35726	1.60094	1.67262	1.67974	1.67981	1.67981

(II) Newton-Quasi algorithm Stage II

Iteration	1	2	3	4	5
$\hat{\mu}_v^* = \hat{\mu}_B^* = t^*$	3.6921146	3.7070019	3.7071552	3.7071552	3.7071552
F	417.49514	416.86883	416.86299	416.86299	416.86299
$\partial F/\partial t$	-8.2040178	-0.7069619	-0.0063172	-5×10^{-7}	3×10^{-13}
L_s	1.5913926	1.5084666	1.5004011	1.5004011	1.5004011
$E[L_sNW]$	0.5156125	0.4774836	0.4738062	0.4737730	0.4737729
$E[WV]$	2.4843875	2.5225164	2.5261938	2.5262271	2.5262271
$E[L_sWV]$	0.8937283	0.8767539	0.8749862	0.8749701	0.8749701
$E[IDWV]$	1.5906592	1.6457625	1.6512076	1.6512569	1.6512569

Next, we investigate the effect of system parameters on the optimal solution and the system performance measures. For various values of λ and η, some numerical results of the optimal solution and the corresponding system performance measures are shown in Table 3. From Table 3, it is observed that (i) F increases when λ or η become larger; because the raise of average arrival rate or average vacation rate would increase the queueing length and the number of customers in the system; (ii) from the first three columns, s^* is insensitive to the parameter η; (iii) the optimal value s^* increases when the arrival rate becomes larger; that is, more tunable wavelengths are required when the traffic intensity grows up; and (iv) μ_v^* and μ_B^* becomes larger as λ increases which is a reasonable result.

7. **Concluding remarks.** The infinite capacity M/M/s queue with multiple working vacations is studied (denoted by M/M/s/WV). The matrix geometric method works well for computing the steady-state probabilities in this paper. This paper generalizes the M/M/1/WV queue. Moreover, an efficient algorithm (Newton-Quasi algorithm) is developed for searching the approximate optimum values $(\hat{s}^*, \hat{\mu}_v^*, \hat{\mu}_B^*)$ that minimize the cost function.

TABLE 3. System performance measures of an M/M/s queue with multiple working vacations under optimal operating conditions.

(λ, η)	(5.0, 0.3)	(5.0, 0.6)	(5.0, 0.9)	(2.5, 0.5)	(5.0, 0.5)	(7.5, 0.5)
$\hat{s}^*$	3	3	3	2	3	3
$\hat{\mu}_v^*$	3.57169	3.73321	3.12192	2.79076	3.70716	4.75075
$\hat{\mu}_B^*$	3.57169	3.76697	4.30445	2.79076	3.70716	4.75075
F	399.313	423.440	436.139	300.583	416.863	503.624
L_s	1.57691	1.47988	1.53670	1.12063	1.50040	1.87384
$E[L_sNW]$	0.37561	0.51109	0.60014	0.31588	0.47377	0.51854
$E[WV]$	2.62439	2.48891	2.39986	1.68412	2.52623	2.48146
$E[L_sWV]$	1.02429	0.82362	0.77411	0.57993	0.87497	1.06016
$E[IDWV]$	1.60010	1.66529	1.62575	1.10419	1.65126	1.42130

Acknowledgments. The authors thank the referees for their remarks and comments, which helped us to improve the clarity of the article.

REFERENCES

[1] Y. Baba, *Analysis of a GI/M/1 queue with multiple working vacations*, Operations Research Letters, **33** (2005), 201–209.

[2] A. D. Banik, U. C. Gupta and S. S. Pathak, *On the GI/M/1/N queue with multiple working vacations-analytic analysis and computation*, Applied Mathematical Modelling, **31** (2006), 1701–1710.

[3] R. L. Burden and J. Douglas, "Numerical Analysis," 7^{th} Edition, Brooks/Cole, USA, 2001.

[4] U. Chatterjee and S. P. Mukherjee, *GI/M/1 queue with server vacations*, Journal of the Operational Research Society, **41** (1990), 83–87.

[5] E. K. P. Chong and S. H. Zak, "An Introduction to Optimization," 2^{nd} Edition, Wiley, New York, 2001.

[6] B. T. Doshi, *Queueing systems with vacations–a survey*, Queueing Systems Theory Appl., **1** (1986), 29–66.

[7] S. W. Fuhrmann and R. B. Cooper, *Stochastic decompositions in the M/G/1 queue with generalized vacations*, Operations Research, **33** (1985), 1117–1129.

[8] F. Karaesmen and S. M. Gupta, *The finite capacity GI/M/1 queue with server vacations*, Journal of the Operational Research Society, **47** (1996), 817–828.

[9] T. Lee, *The M/G/1/N queue with vacation and exhaustive service discipline*, Operations Research, **32** (1984), 774–784.

[10] J.-H. Li, W.-Y. Liu and N.-S. Tian, *Discrete time GI/Geo/1 queue with multiple working vacations* Queueing Systems, **56** (2007), 53–63.

[11] C.-H. Lin and J.-C. Ke, *Multi-server system with single working vacation*, Applied Mathematical Modelling, **33** (2009), 2967–2977.

[12] W.-Y. Liu, X.-L. Xu and N.-S. Tian, *Stochastic decomposition in the M/M/1 queue with working vacations*, Operations Research Letters, **35** (2007), 595–600.

[13] M. F. Neuts, "Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach," Johns Hopkins Series in the Mathematical Sciences, **2**, Johns Hopkins University Press, Baltimore, Md., 1981.

[14] L. D. Servi and S. G. Finn, *M/M/1 queues with working vacations (M/M/1/WV)*, Performance Evaluation, **50** (2002), 41–52.

[15] H. Takagi, "Queueing Analysis: A Foundation of Performance Evaluation," Vol. 1, Vacation and Priority Systems, Part 1, North-Holland Publishing Co., Amsterdam, 1991.

[16] N. Tian, D. Zhang and C. Cao, *The GI/M/1 queue with exponential vacations*, Queueing Systems Theory Appl., **5** (1989), 331–344.

[17] J. A. White, J. W. Schmidt and G. K. Benett, "Analysis of Queueing System," Operations Research and Industrial Engineering, Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1975.

[18] D. A. Wu and H. Takagi, *M/G/1 queue with multiple working vacations*, Performance Evaluation, **63** (2006), 654–681.

Received December 2009; 1st revision July 2010; final revision June 2011.

E-mail address: duckboy614583@gmail.com
E-mail address: khwang@asia.edu.tw
E-mail address: jauchuan@ntit.edu.tw
E-mail address: jbke@amath.nchu.edu.tw