Computers & Industrial Engineering 62 (2012) 296–305

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Multi-server machine repair model with standbys and synchronous multiple vacation

Jau-Chuan Ke [a,*], Chia-Huang Wu [b]

[a] Department of Applied Statistics, National Taichung Institute of Technology, Taichung, Taiwan, ROC
[b] Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan, ROC

ARTICLE INFO

Article history:
Received 1 April 2011
Received in revised form 23 September 2011
Accepted 24 September 2011
Available online 2 October 2011

Keywords:
Machine repair models
Synchronous multiple vacation
Standby

ABSTRACT

This paper investigates a machine repair problem with homogeneous machines and standbys available, in which multiple technicians are responsible for supervising these machines and operate a (R, V, K) synchronous vacation policy. With such a policy, if any V idle technicians exist in the system, these V $(V < R)$ technicians would take a synchronous vacation. Upon returning from vacation, they would take another vacation if there is no broken machine waiting in the queue. This pattern continues until at least one failed machine arrives. It is assumed that the number of teams/groups on vacation is less than or equal to K $(0 \leq KV < R)$. The matrix analytical method is employed to obtain a steady-state probability and the closed-form expression of the system performance measures. Efficient approaches are performed to deal with the optimization problem of the discrete/continuous variables while maintaining the system availability at a specified acceptable level.

1. Introduction

In many industrial processes, production machines are unreliable and may have a breakdown. When a machine fails, it is sent to a maintenance facility and repaired by a group of technicians (servers). In order to achieve the production quota and reduce the loss of production capacity, the plant usually keeps standby machines that could substitute for a failed machine. In this paper, a machine repair problem, which includes M identical machines, S standby machines, and R technicians with synchronous multiple vacation policy is investigated. There are numerous researches on the machine repair problem or the multi-server queueing system with various vacation policies.

1.1. Machine repair/inference problem

This paper first conducts a literature review on non-vacation servers (*i.e.* servers do not perform secondary tasks during their idle period). Ke and Wang (1999) analyzed machine repair problems with constant balking probability, negative exponential distributed reneging, and unreliable servers. A subsequent study, by Ke and Wang (1999) and Wang and Ke (2003), revisited this model with reneging behavior. The system steady-state availability, MTTF, and some system performance measures were presented. Wang, Ke, and Ke (2007) investigated the profit analysis of the M/M/R machine repair problem with balking, reneging, and standby switching failures. They employed the direct search method and the steepest descent method in order to determine global maximum values to satisfy system constraints. A comprehensive and exhaustive discussion of machine repair problems was given by Haque and Armstrong (2007). Ke and Lin (2008) modeled manufacturing systems using two queueing systems with different repair rates and different numbers of technicians. As for vacation servers, Gupta (1997) first investigated a machine interference problem with warm spares and server vacations, including multiple vacations, single vacation, and hybrid multiple/single vacation schemes. A transform free, closed form expression of the probability distribution for the number of operating machines and performance measures was developed in Gupta's work. Ke (2006) generalized Gupta's work to unreliable-server cases. Numerical investigation and sensitivity analysis of the reliability and availability measures of a repair system were investigated by Ke and Lin (2005), in which the servers were imperfect, and applied a multiple vacation policy. Ke and Wang (2007) dealt with machine repair problems with a single/multiple vacation policy and two type spares, and a cost analysis for both vacation models was developed. Recently, Wang, Chen, and Yang (2009) studied the M/M/1 machine repair problem with a working vacation policy, where the server may work with different repair rates rather than completely terminate during a vacation period. In their work, the

* Corresponding author. Address: Department of Applied Statistics, National Taichung Institute of Technology, No. 129, Sec. 3, Sanmin Rd., Taichung 404, Taiwan, ROC. Tel.: +886 4 22196330; fax: +886 4 22196331.
E-mail addresses: jauchuan@ntit.edu.tw (J.-C. Ke), duckboy614583@gmail.com (C.-H. Wu).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.09.017

optimal number of machines and two different repair rates were determined using a direct search method and Newton's method.

1.2. Queueing model with synchronous vacation policy

The second category of studies is in regard to a queueing system with synchronous vacation policy. Zhang and Tian (2003a, 2003b) first introduced the multi-server queueing system with single/multiple synchronous vacations, in which some idle servers would take a vacation of random duration when finished serving customers, with no customers waiting. Moreover, Tian and Zhang (2003) investigated a GI/M/c queueing system with phase-type vacations where all servers take multiple vacations together until the system is not empty. Tian and Zhang (2006) considered a multi-server queueing system with a (d, N) vacation policy, in which d idle servers may take multiple vacations together until the number is equal to or more than a predetermined threshold N. They also conducted a computational study of the optimal value of controllable variable d, while N was presented under non-controllable parameter. Yue et al. (2006) studied a finite capacity queueing system with balking, reneging, and single synchronous vacations policy. They also obtained a matrix-form solution for the steady-state probability vector and some performance measures. A multi-server queueing model, with Markovian arrivals and synchronous phase type vacations, was investigated by Srinivas (2007), who performed several cases of MAP processes and numerical examples, including the tables of optimum values of system parameters, the corresponding system performances, and total expected costs were presented. Recently, Srinivas (2009) presented a steady-state analysis of the MAP/M/c queueing system with the phase type vacation and provided some interesting numerical results. However, existing research works regarding synchronous vacation do not include machine repair problems, and mainly focused on the infinite capacity queueing system.

In the photolithography process (see Uzsoy, Lee, & Martin-Vega, 1992, 1994), each job is processed by stepper machines, which are unreliable and are subject to unpredictable breakage. When a machine fails it is immediately sent to the maintenance department and repaired by technicians, as the stepper machines are critical resources in the photolithography process, thus, maintaining the machines operational performance in the system is very important. In the repair facility, an arriving broken stepper machine undergoes a random process. The service/repair time of each failed machine is by provided a technician, and could be regarded as a random variable. For the convenience of labor management, the technicians usually are divided into teams/groups of fixed size. Whenever there is an idle team, they would take a multiple vacation and leave the repair facility at random periods. The primary goal of leaving the repair facility is to improve the utility of the work force (support for other departments), or increase the abilities of personnel by joining a training course. A broken machine must wait for repair service in a queue when there is no available technician/server in the system. Therefore, the plant always maintains some standby machines to substitute the failed machines. As mentioned above, to allocate labor and maintain the operations of product machines are important for the engineers and management. However, regarding production or manufacturing systems, there are no studies on partial server multiple vacation (synchronous vacation) policies for the machine repair problems or finite arrival resources.

The objectives of this paper are as follows: (1) provide a matrix-analytical computational algorithm to develop the steady-state probability vectors; (2) derive the steady-state availability, the mean time to failure (MTTF), and other system performance measures; (3) construct a cost model to determine the optimal number of technicians (servers), the optimal vacation policy, the optimal service rate, and the optimal vacation rate; (4) conduct numerical study on the effect of parameters on the system characteristics.

2. The system

This paper considers a multi-server machine repair problem with a synchronous multiple vacation policy and standby. There are M operating machines, S standby machines, and R technicians (servers) in this system. The presented machine repair system with warm standbys is shown in Fig. 1. A repaired machine would stay as a standby (Case (I)) or be returned to the product line if the system is short (Case (II)).

The detailed descriptions and assumptions of this model are given as follows:

1. M operating machines are required for the function of the system. In other words, the system is short only if $S + 1$ (or more) machines fail.
2. Operating machines are subject to breakdowns, according to an independent Poisson process, with rate λ. When an operating machine breaks down, it is immediately backed up by an available standby.
3. Each of the standby machines fails independently of the others with Poisson rate α, where ($0 \leqslant \alpha \leqslant \lambda$). When a standby machine moves into an operating state, its characteristics are the same as an operating machine.
4. Failed machines in the system form a single waiting line and receive repair in the order of their breakdown, *i.e.* FCFS discipline. The service time provided by each technician is an independent and identically distributed exponential random variable with rate μ.

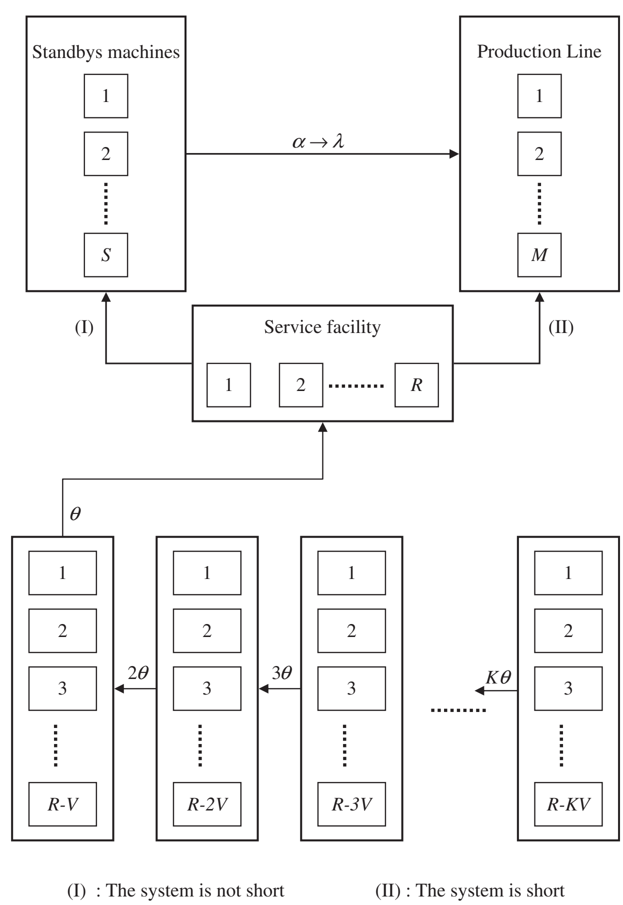

Fig. 1. A machine repair problem with standby machines.

5. When a failed machine is repaired, it enters into a standby state unless the system is short, then the repaired machine would be sent back to an operating state.
6. Each technician could repair only one failed machine at a time, and a failed machine arriving at the repair facility where all technicians are busy or on vacation must wait in the queue until a technician is available.
7. When there are any V idle technicians, they take a synchronous multiple vacation. Upon returning from the vacation, they would take a vacation again if there are no fail machines waiting in the queue. The number of teams/groups on synchronous vacation is restricted no more than K $(1 \leqslant K \leqslant [R/V] - 1)$ at any time. The symbol "$[\cdot]$" is the ceiling function which $[x]$ denotes the smallest integer not less than x.
8. The vacation time of each team/group has an exponential distribution with parameter θ. The various stochastic processes involved in this system are independent of each other.

It should be noted that the inequality equation, $1 \leqslant K \leqslant [R/V] - 1$, means that it is not allowed to have all technicians (servers) on vacation at any time. In other words, the constraint $KV < R$ must hold surely. Therefore, the vacation policy introduced by this study is a vacation policy without exhausting the servers, which is different from the vacation polices in literatures, but closer to practical use than past studies. The vacation policy we mentioned above is represented by "(R, V, K) synchronous multiple vacation policy" and could be used to expresses a queueing system with R servers and K teams/groups (with size V) are allowed to take synchronous vacation.

3. Steady-state results

For the multi-server machine repair model, with a (R, V, K) synchronous multiple vacation policy and standby machines, the state of the system could be described by the pairs $\{(i, n): i = R, R - V, R - 2V, \ldots, R - KV,$ and $n = \max\{i - V + 1, 0\}, \ldots, M + S\}$, where i denotes the number of operating (not on vacation) technicians in the system, and n represents the number of failed machines in the system. For instance, (3, 2) indicates the state of 3 operating machines ($R - 3$ machines are on vacation) and 2 failed machines in the system. The mean failure rate λ_n and mean repair rate μ_n for this system are given by:

$$\lambda_n = \begin{cases} M\lambda + (S - n)\alpha, & 0 \leqslant n \leqslant S \\ [M - (n - S)]\lambda, & S \leqslant n \leqslant M + S \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mu_n = \begin{cases} n\mu, & 1 \leqslant n \leqslant R \\ 0, & \text{otherwise} \end{cases}$$

In the steady-state, the following notations are used:

$P_{i,n} \equiv$ probability that there are n failed machines in the system when there are i operating technicians in the system ($R - i$ technicians are on vacation), where $i = R, R - V, \ldots, R - (K - 1)V, R - KV$, $n = \max\{i - V + 1, 0\}, \ldots, M + S$.

3.1. Steady-state equations

In reference to the transition diagram shown in Fig. 2, the steady-state equations for multiple-server machine repair problems, with standby under a (R, V, K) synchronous multiple vacation policy, are obtained as follows.

(1) $i = R - KV$

$$\lambda_0 P_{R-KV,0} = \mu_1 P_{R-KV,1} \tag{1}$$

$$(\lambda_n + \mu_n) P_{R-KV,n} = \lambda_{n-1} P_{R-KV,n-1} + \mu_{n+1} P_{R-KV,n+1}, \quad 1 \leqslant n \leqslant R - KV - 1 \tag{2}$$

$$(\lambda_{R-V} + \mu_{R-V}) P_{R-KV,R-V} = \lambda_{R-KV-1} P_{R-KV,R-V-1} + \mu_{R-KV} P_{R-KV,R-KV+1} + \mu_{R-KV+1} P_{R-(K-1)V,R-KV+1} \tag{3}$$

$$(\lambda_n + \mu_n + K\theta) P_{R-KV,n} = \lambda_{n-1} P_{R-KV,n-1} + \mu_{R-KV} P_{R-KV,n+1}, \quad R - KV + 1 \leqslant n \leqslant M + S - 1 \tag{4}$$

$$(\mu_{R-KV} + K\theta) P_{R-KV,M+S} = \lambda_{M+S-1} P_{R-KV,M+S-1} \tag{5}$$

(2) $R - (K-1)V \leqslant i \leqslant R - V$

$$(\lambda_{R-(i+1)V+1} + \mu_{R-(i+1)V+1}) P_{R-iV,R-(i+1)V+1} = (i+1)\theta P_{R-(i+1)V,R-(i+1)V+1} + \mu_{R-(i+1)V+2} P_{R-iV,R-(i+1)V+2} \tag{6}$$

$$(\lambda_n + \mu_n) P_{R-iV,n} = (i+1)\theta P_{R-(i+1)V,n} + \lambda_{n-1} P_{R-iV,n-1} + \mu_{n+1} P_{R-iV,n+1}, \quad R - (i+1)V + 2 \leqslant n \leqslant R - iV - 1 \tag{7}$$

$$(\lambda_{R-iV} + \mu_{R-iV}) P_{R-iV,R-iV} = (i+1)\theta P_{R-(i+1)V,R-iV} + \lambda_{R-iV-1} P_{R-iV,R-iV-1} + \mu_{R-iV} P_{R-iV,R-iV+1} + \mu_{R-iV+1} P_{R-(i-1)V,R-iV+1} \tag{8}$$

$$(\lambda_n + \mu_{R-iV} + i\theta) P_{R-iV,n} = (i+1)\theta P_{R-(i+1)V,n} + \lambda_{n-1} P_{R-iV,n-1} + \mu_{R-iV} P_{R-iV,n+1} R - iV + 1 \leqslant n \leqslant M + S - 1 \tag{9}$$

$$(\mu_{R-iV} + i\theta) P_{R-iV,M+S} = (i+1)\theta P_{R-(i+1)V,M+S} + \lambda_{M+S-1} P_{R-iV,M+S-1} \tag{10}$$

(3) $i = R$

$$(\lambda_{R-V+1} + \mu_{R-V+1}) P_{R,R-V+1} = \theta P_{R-V,R-V+1} + \mu_{R-V+2} P_{R,R-V+2} \tag{11}$$

$$(\lambda_n + \mu_n) P_{R,n} = \theta P_{R-V,n} + \lambda_{n-1} P_{R,n-1} + \mu_{n+1} P_{R,n+1}, \quad R - V + 2 \leqslant n \leqslant R - 1 \tag{12}$$

$$(\lambda_n + \mu_R) P_{R,n} = \theta P_{R-V,n} + \lambda_{n-1} P_{R,n-1} + \mu_R P_{R,n+1}, \quad R \leqslant n \leqslant M + S - 1 \tag{13}$$

$$\mu_R P_{R,M+S} = \theta P_{R-V,M+S} + \lambda_{M+S-1} P_{R,M+S-1} \tag{14}$$

There is no way of solving (1)–(14) in a recursive manner in order to develop explicit expressions for steady-state probabilities. In the next section, this study provides a matrix-analytic method to address this problem.

3.2. Matrix-analytical solutions

To analyze the resulting system of linear Eqs. (1)–(14), a matrix-analytic approach is used. Following concepts by Neuts (1981), in order to represent the steady-state equations in a matrix-form, the transition rate matrix **Q** (the coefficient matrix) of this Markov chain could be partitioned as follows:

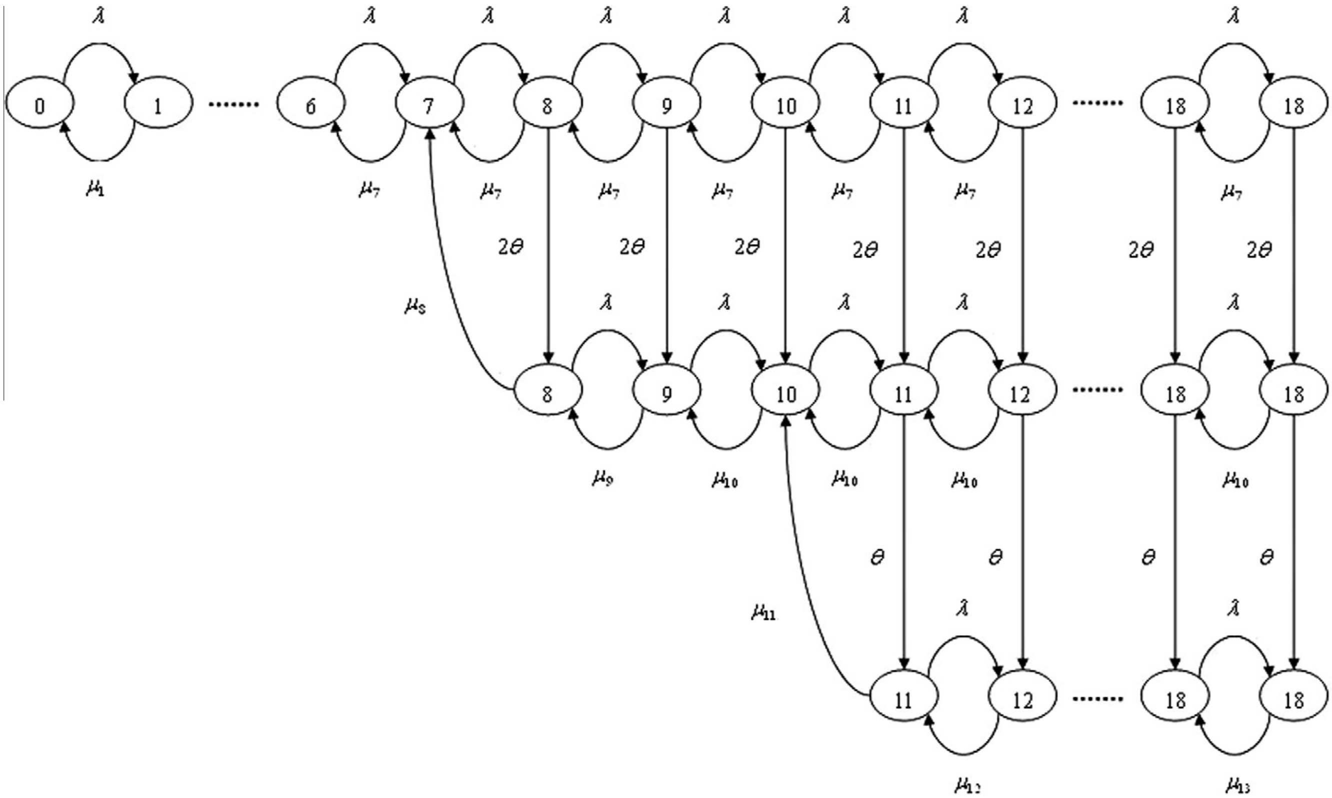

Fig. 2. Steady-transition-rate diagram for a multi-server machine repair problem with standbys under a (R, V, K) synchronous multiple vacation policy ($M = 15, S = 3, R = 13, V = 3, K = 2$).

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A}_K & \mathbf{B}_{K-1} & & & & & \\ \mathbf{C}_K & \mathbf{A}_{K-1} & \mathbf{B}_{K-2} & & & & \\ & \mathbf{C}_{K-1} & \mathbf{A}_{K-2} & \mathbf{B}_{K-3} & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{C}_3 & \mathbf{A}_2 & \mathbf{B}_1 & & \\ & & & \mathbf{C}_2 & \mathbf{A}_1 & \mathbf{B}_0 & \\ & & & & \mathbf{C}_1 & \mathbf{A}_0 & \end{pmatrix} \tag{15}$$

Matrix $\mathbf{Q}$ is a square matrix of order $K(M+S-R) - K(K+1)/2 + (M+S+1)$, and each entry of the matrix $\mathbf{Q}$ is listed in the following. The first diagonal sub-matrix is

$$\mathbf{A}_K = \begin{pmatrix} * & \lambda_0 & & & & & \\ \mu_1 & * & \lambda_1 & & & & \\ & \mu_2 & * & \lambda_2 & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mu_{R-KV} & * & \lambda_{R-KV} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mu_{R-KV} & * & \lambda_{M+S-1} \\ & & & & & \mu_{R-KV} & * \end{pmatrix}_{(M+S+1)\times(M+S+1)} \tag{16}$$

$\mathbf{A}_K$ describes the in-flows and out-flows of the states in the first (top) level as Fig. 2. Later, for $i = 1, 2, \ldots, K-1$,

$$\mathbf{A}_i = \begin{pmatrix} * & \lambda_{R-(i+1)V+1} & & & & & \\ \mu_{R-(i+1)V+2} & * & \lambda_{R-(i+1)V+2} & & & & \\ & \mu_{R-(i+1)V+3} & * & \lambda_{R-(i+1)V+3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mu_{R-iV} & * & \lambda_{R-iV} & \\ & & & & \ddots & \ddots & \ddots \\ & & & & \mu_{R-iV} & * & \lambda_{M+S-1} \\ & & & & & \mu_{R-iV} & * \end{pmatrix}_{(M+S-R+(i+1)V)\times(M+S-R+(i+1)V)} \tag{17}$$

$\mathbf{A}_i$ record the corresponding flows in the middle levels. Finally, the matrix $\mathbf{A}_0$ as following shows the flows in the bottom (lowest) level as Fig. 2.

$$\mathbf{A}_0 = \begin{pmatrix} * & \lambda_{R-V+1} & & & & & \\ \mu_{R-V+2} & * & \lambda_{R-V+2} & & & & \\ & \mu_{R-V+3} & * & \lambda_{R-V+3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mu_R & * & \lambda_R & \\ & & & & \ddots & \ddots & \ddots \\ & & & & \mu_R & * & \lambda_{M+S-1} \\ & & & & & \mu_R & * \end{pmatrix}_{(M+S-R+V)\times(M+S-R+V)} \tag{18}$$

The diagonal elements of matrix $\mathbf{A}_i$ ($\mathbf{Q}$), indicated by $*$, are such that the sum of each row of $\mathbf{Q}$ is zero (keeps the flow balance).

Then, the first lower-diagonal sub-matrices, $\mathbf{C}_K$ is a matrix of size $(M+S-R+KV) \times (M+S+1)$ with only one nonzero element $\mathbf{C}_K[1, R-KV+1] = \mu_{R-KV+1}$. For $i = 1, 2, \ldots, K-1$, $\mathbf{C}_i$ is a matrix of size $(M+S-R+iV)\times[M+S-R+(i+1)V]$ with only one nonzero element $\mathbf{C}_i[1,V]=\mu_{R-iV+1}$.

$\mathbf{B}_{K-1}$ is a matrix of size $(M+S+1)\times(M+S-R+KV)$ with elements $\mathbf{B}_{K-1}[R+1-KV+n,n]=K\theta$, $n=1,2,\ldots,(M+S-R+KV)$. For $i=0,1,\ldots,K-2$, $\mathbf{B}_i$ is a matrix of size $[M+S-R+(i+2)V]\times[M+S-R+(i+1)V]$ with elements $\mathbf{B}_i[V+n,n]=(i+1)\theta$, $n=1,2,\ldots,M+S-R+(i+1)V$.

For example, for the queueing system in Fig. 2 ($M=15, S=3, R=13, V=3, K=2$), $\mathbf{C}_2$ (11×19) and $\mathbf{C}_1$ (size 8×11) have one and only one nonzero element $\mathbf{C}_2[1,8]=\mu_8$ and $\mathbf{C}_1[1,3]=\mu_{11}$, respectively. $\mathbf{B}_1$ (19×11) has nonzero elements $\mathbf{B}_1[9,1],\mathbf{B}_1[10,2],\ldots,\mathbf{B}_1[18,10]$ and $\mathbf{B}_1[19,11]$ with value 2θ. Similarly, $\mathbf{B}_0$ (11×8) has nonzero elements $\mathbf{B}_0[4,1],\mathbf{B}_0[5,2],\ldots,\mathbf{B}_0[10,7]$ and $\mathbf{B}_0[11,8]$ with value θ.

Let $\mathbf{\Pi}$ denote the steady-state probability vector of $\mathbf{Q}$. Vector $\mathbf{\Pi}$ is partitioned as $\mathbf{\Pi}=[\mathbf{\Pi}_K,\mathbf{\Pi}_{K-1},\ldots,\mathbf{\Pi}_1,\mathbf{\Pi}_0]$ where $\mathbf{\Pi}_K=[P_{R-KV,0},P_{R-KV,1},\ldots,P_{R-KV,M+S-1},P_{R-KV,M+S}]$ denotes the steady-state probability vector that the number of teams/groups on vacation is equal to K (such as the states of the top level in Fig. 2). The sub-vectors $\mathbf{\Pi}_k=[P_{R-kV,\,R-(k+1)V+1},P_{R-kV,\,R-(k+1)V+2},\ldots,P_{R-kV,M+S}]$ represents the steady-state probability vector that the number of teams/group on vacation is equal to k, $k=0,1,\ldots,K-1$. Then, the steady-state equations are expressed in matrix-form as $\mathbf{\Pi Q}=\mathbf{0}$ are given by

$$\mathbf{\Pi}_K\mathbf{A}_K+\mathbf{\Pi}_{K-1}\mathbf{C}_K=\mathbf{0} \tag{19}$$

$$\mathbf{\Pi}_{k+1}\mathbf{B}_k+\mathbf{\Pi}_k\mathbf{A}_k+\mathbf{\Pi}_{k-1}\mathbf{C}_k=\mathbf{0},\quad k=1,2,\ldots,K-1 \tag{20}$$

$$\mathbf{\Pi}_1\mathbf{B}_0+\mathbf{\Pi}_0\mathbf{A}_0=\mathbf{0} \tag{21}$$

and the following normalizing equation must be satisfied

$$\sum_i\sum_n P_{i,n}=\sum_k\mathbf{\Pi}_k\mathbf{e}=1 \tag{22}$$

where $\mathbf{e}$ represents a column vector with suitable size and each component equal to one. Firstly, it is known that $\mathbf{\Pi}_K=\mathbf{\Pi}_{K-1}\mathbf{C}_K(-\mathbf{A}_K)^{-1}=\mathbf{\Pi}_{K-1}\phi_K$ from Eq. (19). Substituting this result to Eq. (20) and performing some routine manipulations, we obtain the following results

$$\mathbf{\Pi}_K\mathbf{B}_{K-1}+\mathbf{\Pi}_{K-1}\mathbf{A}_{K-1}+\mathbf{\Pi}_{K-2}\mathbf{C}_{K-1}=\mathbf{\Pi}_{K-1}\phi_K\mathbf{B}_{K-1}+\mathbf{\Pi}_{K-1}\mathbf{A}_{K-1}$$
$$+\mathbf{\Pi}_{K-2}\mathbf{C}_{K-1}=\mathbf{\Pi}_{K-1}[\phi_K\mathbf{B}_{K-1}+\mathbf{A}_{K-1}]+\mathbf{\Pi}_{K-2}\mathbf{C}_{K-1}=\mathbf{0}$$

implies

$$\mathbf{\Pi}_{K-1}=\mathbf{\Pi}_{K-2}\mathbf{C}_{K-1}[-(\phi_K\mathbf{B}_{K-1}+\mathbf{A}_{K-1})]^{-1}=\mathbf{\Pi}_{K-2}\phi_{K-1}$$

Similarly, doing these recursive operations successively, we have

$$\mathbf{\Pi}_k=\mathbf{\Pi}_{k-1}\mathbf{C}_k\left[-(\phi_{k+1}\mathbf{B}_k+\mathbf{A}_k)\right]^{-1}=\mathbf{\Pi}_{k-1}\phi_k,\ 1\leqslant k\leqslant K-1 \tag{23}$$

where $\phi_k=\mathbf{C}_k[-(\phi_{k+1}\mathbf{B}_k+\mathbf{A}_k)]^{-1}$, $1\leqslant k\leqslant K-1$. Finally, replacing $\mathbf{\Pi}_1$ by $\mathbf{\Pi}_0\phi_1$ in Eq. (21) infers

$$\mathbf{\Pi}_0\phi_1\mathbf{B}_0+\mathbf{\Pi}_0\mathbf{A}_0=\mathbf{\Pi}_0(\phi_1\mathbf{B}_0+\mathbf{A}_0)=\mathbf{0} \tag{24}$$

Consequently, $\mathbf{\Pi}_k(1\leqslant k\leqslant K)$ in Eqs. (19) and (20) could be written in terms of $\mathbf{\Pi}_0$ as $\mathbf{\Pi}_k=\mathbf{\Pi}_0\mathbf{\Phi}_k$ where $\mathbf{\Phi}_k=\phi_1,\phi_2,\ldots,\phi_k$, $1\leqslant k\leqslant K$ is a product form of quantity ϕ_k. Once the steady-state probability $\mathbf{\Pi}_0$ being obtained, the steady-state solutions $\mathbf{\Pi}=[\mathbf{\Pi}_K,\mathbf{\Pi}_{K-1},\ldots,\mathbf{\Pi}_1,\mathbf{\Pi}_0]$ are then determined. $\mathbf{\Pi}_0$ could be solved by Eq. (24), with the following normalization equation

$$\sum_i\sum_n P_{i,n}=\sum_k\mathbf{\Pi}_k\mathbf{e}=\mathbf{\Pi}_0\left[\sum_{k=1}^{K}\mathbf{\Phi}_k+\mathbf{I}\right]\mathbf{e}=1 \tag{25}$$

where $\mathbf{I}$ represents an identity matrix with suitable size. Solving Eq. (24) and the above normalization condition simultaneously would gain the steady-state solution $\mathbf{\Pi}_0$.

4. Performance analysis

This section addresses the steady-state availability and the mean time to system failure analysis. In addition, the explicit expressions of other system performance measures in the system are obtained.

4.1. Availability and reliability analysis

It is noted that the system fails if, and only if, $S+1$ (or more) machines fail. Hence, the steady-state availability could be calculated as

$$A.V.=P(0\leqslant n\leqslant S)=\sum_i\sum_{0\leqslant n\leqslant S}P_{i,n} \tag{26}$$

The mean time to failure (MTTF) is an important reliability characteristic, as a shortage in the system (system failure) would cause operating costs to be raised and abate the utility of the production machines. To calculate the MTTF, the original transition rate matrix is reduced by deleting the rows and columns of the absorbing state(s). The new reduced transient matrix, $\mathbf{\Gamma}$, is as follows:

$$\mathbf{\Gamma}=\begin{pmatrix}\mathbf{A}_K^r & \mathbf{B}_{K-1}^r & & & & & \\ \mathbf{C}_K^r & \mathbf{A}_{K-1}^r & \mathbf{B}_{K-2}^r & & & & \\ & \mathbf{C}_{K-1}^r & \mathbf{A}_{K-2}^r & \mathbf{B}_{K-3}^r & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mathbf{C}_3^r & \mathbf{A}_2^r & \mathbf{B}_1^r & & \\ & & & \mathbf{C}_2^r & \mathbf{A}_1^r & \mathbf{B}_0^r \\ & & & & \mathbf{C}_1^r & \mathbf{A}_0^r\end{pmatrix} \tag{27}$$

where $\mathbf{A}_i^r$, $\mathbf{B}_i^r$, and $\mathbf{C}_i^r$ denote the sub-matrices of $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$, respectively. These sub-matrices $\mathbf{A}_i^r$, $\mathbf{B}_i^r$, and $\mathbf{C}_i^r$, with superscript "r", are derived by deleting the rows and columns of the absorbing state(s) ((i,n), which satisfy $n\geqslant S+2$) from the original matrices $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$. Then, the expected time to reach an absorbing state could be calculated from

$$E\left[T_{\mathbf{P}(0)\to\mathbf{P}_{(\text{absorbing})}}\right]=\mathbf{P}(0)^{\mathrm{T}}\int_0^\infty e^{\Gamma t}\,dt=\mathbf{P}(0)^{\mathrm{T}}(-\mathbf{\Gamma}^{-1})\mathbf{e} \tag{28}$$

where $\mathbf{P}(0)=[1,0,\ldots,0]^{\mathrm{T}}$ denotes the initial conditions for this problem (see Wang & Ke, 2003).

4.2. System performance measures

The analysis of this study is based on the following system performance measures. Let

$E[F]\equiv$ the expected number of failed machines in the system,
$E[F_q]\equiv$ the expected number of failed machines in the queue,
$E[O]\equiv$ the expected number of operating machines in the system,
$E[S]\equiv$ the expected number of acting standby machines in the system,
$E[B]\equiv$ the expected number of busy repairmen in the system,
$E[V]\equiv$ the expected number of vacation repairmen in the system,
$E[I]\equiv$ the expected number of idle repairmen in the system,
$M.A.\equiv$ machine availability (the fraction of the total time that the machines are working),
$O.U.\equiv$ operative utilization (the fraction of busy servers).

For convenience, this study defines the symbol "$\langle f(n), n=a..b\rangle$" as denoting a column vector with dimension $(b-a+1)$, of which the nth element is $f(n)$. For instance, $\langle$max

$(n,4),4), n = 1..6\rangle = [4,4,4,4,5,6]^T$, $\langle \min\{n,3\}, n = 1..6\rangle = [1,2,3,3,3,3]^T$, and $\langle n^2, n = 1..6\rangle = [1,4,9,16,25,36]^T$. It represents a column vector that is constructed by using the specific function. Then, the expressions for $E[F]$, $E[F_q]$, $E[O]$, $E[S]$, $E[B]$, $E[V]$, and $E[I]$ are developed as follows:

$$E[F] = \sum_i \sum_n nP_{i,n} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle n, n = [R-(k+1)V]..(M+S)\rangle + \boldsymbol{\Pi}_K \langle n, n = 0..(M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle n, n = [R-(k+1)V] \cdot (M+S)\rangle + \boldsymbol{\Phi}_K \langle n, n = 0..(M+S)\rangle \right\} \quad (29)$$

$$E[F_q] = \sum_i \sum_n P_{i,n} \max\{n-i,0\} + \sum_n P_{R-KV,n} \max\{n-i,0\} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle \max\{n-(R-kV),0\}, n=[R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Pi}_K \langle \max\{n-(R-kV),0\}, n=0..(M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle \max\{n-(R-kV),0\}, n=[R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Phi}_K \langle \max\{n-(R-KV),0\}, n=0..(M+S)\rangle \right\} \quad (30)$$

$$E[O] = \sum_i \sum_n P_{i,n} \min\{M, M+S-n\} + \sum_n P_{R-KV,n} \min\{M, M+S-n\} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle \min\{M, M+S-n\}, n=[R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Pi}_K \langle \min\{M, M+S-n\}, n=0..(M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle \min\{M, M+S-n\}, n=[R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Phi}_K \langle \min\{M, M+S-n\}, n=n=0..(M+S)\rangle \right\} \quad (31)$$

$$E[S] = \sum_i \sum_n P_{i,n} \max\{0, S-n\} + \sum_n P_{R-KV,n} \max\{0, S-n\} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle \max\{0, S-n\}, n = [R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Pi}_K \langle \max\{0, S-n\}, n = 0..(M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle \max\{0, S-n\}, n = [R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Phi}_K \langle \max\{0, S-n\}, n = n = 0..(M+S)\rangle \right\} \quad (32)$$

$$E[V] = \sum_{k=1}^{K} KV \boldsymbol{\Pi}_k \mathbf{e} = \boldsymbol{\Pi}_0 V \sum_{k=1}^{K} k \boldsymbol{\Phi}_k \mathbf{e} \quad (33)$$

$$E[I] = \sum_i \sum_n P_{i,n} \max\{i-n, 0\} + \sum_n P_{R-KV,n} \max\{R-KV, 0\} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle \max\{(R-kV)-n, 0\}, n = [R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Pi}_K \langle \max\{(R-KV)-n, 0\}, n = 0..(M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle \max\{(R-kV)-n, 0\}, n = [R-(k+1)V+1]..(M+S)\rangle + \boldsymbol{\Phi}_K \langle \max\{(R-KV)-n, 0\}, n = 0..(M+S)\rangle \right\} \quad (34)$$

$$E[B] = \sum_i \sum_n P_{i,n} \min\{i, n\} = \sum_{k=0}^{K-1} \boldsymbol{\Pi}_k \langle \min\{(R-KV), n\}, n = [R-(k+1)V+1] \cdot (M+S)\rangle + \boldsymbol{\Pi}_K \langle \min\{(R-KV), n\}, n = 0 \cdot (M+S)\rangle = \boldsymbol{\Pi}_0 \left\{ \sum_{k=0}^{K-1} \boldsymbol{\Phi}_k \langle \min\{(R-KV), n\}, n = [R-(k+1)V+1] \cdot (M+S)\rangle + \boldsymbol{\Phi}_K \langle \min\{(R-KV), n\}, n = 0 \cdot (M+S)\rangle \right\} \quad (35)$$

By the properties of minimum and maximum functions, it could be verified that $E[V] + E[I] + E[B] = R$. Furthermore, following Benson and Cox (1951), the machine availability and the operative utilization of servers are defined by

$$M.A. = 1 - \frac{E[F]}{M+S} \quad \text{and} \quad O.U. = \frac{E[B]}{R} \quad (36)$$

Finally, use Little's formula to obtain the expected waiting time in the system, $E[W]$, and in the queue $E[W_q]$, as

$$E[W] = E[F]/\lambda_e \quad \text{and} \quad E[W_q] = E[F_q]/\lambda_e \quad (37)$$

where $\lambda_e = \sum_i \sum_n \lambda_n P_{i,n}$ is the effective arrival rate into the system.

5. Cost analysis

In this section, a total expected cost function per unit time, as based on system performance measures, is constructed. A constraint on system availability is imposed on this cost model, where R, V and K are discrete decision variables. First, let

$C_h \equiv$ cost per unit time when one failed machine joins the system,
$C_e \equiv$ cost per unit time of a failed machine after all standbys are exhaused,
(downtime cost),
$C_s \equiv$ cost per unit time when one machine is functioning as a standby (inventory cost),
$C_b \equiv$ cost per unit time when one repairman is busy,
$C_f \equiv$ cost per unit time of each resident repairman,
$C_t \equiv$ cost per unit time of each team group,
$\gamma \equiv$ cost per unit time of augment the size of team group.

Using the definitions of the cost elements listed above, the total expected cost function per unit time is given by

$$T_{\text{cost}}(R, V, K) = C_h E[F] + C_e(M - E[O]) + C_s E[S] + C_b E[B] + (R - KV)C_f + (R/V)C_t + \gamma V \quad (38)$$

An example (photolithography process problem mentioned in Section 1) is provided to perform the numerical investigation:

- There are $M = 15$ stepper machines and $S = 10$ standby machines in the photolithography process.
- Each operating stepper machine may be interrupted due to unpredictable accidents with Poisson breakdown rate $\lambda = 1.5$.
- The standby machines are with Poisson breakdown rate $\alpha = 1.0$
- In the repair facility, R technicians are responsible to provide the repair service for the failed machines. The repair time for one failed machine is exponentially with mean $\mu^{-1} = 0.2$.
- The servers/technicians are allocated by a (R, V, K) synchronous multiple vacation policy, in which vacation time is an exponential distributed with mean $\theta^{-1} = 2$.
- The cost elements and availability requirements are

$$C_h = 10, \quad C_e = 125, \quad C_s = 90, \quad C_b = 60, \quad C_f = 80, \quad C_t = 45, \quad \gamma = 30, \quad \text{and} \quad A = 0.9$$

The objective of this study is to determine the optimum synchronous multiple vacation policy, including the value of the number of technicians R, say R^*, the team size V, say V^*, and the upper bound of number of vacation teams, say K^*, in order to minimize cost function $T_{\text{cost}}(R, V, K)$ under system availability, is maintained at a certain acceptable level A. Following the concept of Hilliard (1976), this cost minimization problem could be illustrated mathematically as

$$\begin{aligned} &\underset{R,V,K}{\text{Minimize}} \quad T_{\text{cost}}(R,V,K) \\ &\text{Subject to} \quad A.V. \geqslant A \end{aligned} \tag{39}$$

where $A.V.$ (system availability) is the steady-state probability that the number of broken machines is less than or equal to S. The symbol A is an availability level requirement or the acceptable level.

A direct search method may be employed to obtain potentially useful results. An optimization algorithm is a direct search approach over a grid, which boundaries for decision variables are selected in order to guarantee that the global optimum is obtained in the interior region (see Hilliard, 1976). The procedure of direct search method is described in the following.

Algorithm 1. Direct Search Method

INPUT M, S, A and set cost $T^* = \infty$ (initialization)
OUTPUT approximation solution (R^*, V^*, K^*) and $T_{\text{cost}}(R^*, V^*, K^*)$.
Step 1 for $R = 1$ to M do
  Step 2 for $V = 1$ to R do
    Step 3 for $K = 1$ to $[R/V] - 1$ do
    Step 4 Computing $T_{\text{cost}}(R, V, K)$ and the corresponding system availability $A.V.\ (R, V, K)$
    Step 5 If $A.V.\ (R, V, K) \geqslant A$ and $T_{\text{cost}}(R, V, K) \leqslant T^*$
      Step 6 Setting $(R^*, V^*, K^*) = (R, V, K)$ and $T^* = T_{\text{cost}}(R, V, K)$
    Step 7 end if
    Step 8 end do loop
  Step 9 end do loop
Step 10 end do loop
Step 11 OUTPUT (R^*, V^*, K^*), $A.V.$, (R, V, K) and T^*.

The Direct search method using this problem is with complexity $O(M^3)$, and the information of the CPU time spent for computation is listed below in the tables/figures, which are gained by software MAPLE 9, as based on personal computer with implement AMD Dual-Core CPU (2.70 GHz) and 4.0 GB RAM. The computational time is acceptable. The direct search algorithm is applied in the set $\{M = 15 \geqslant R \geqslant V$ and $R > KV$; R,V,K are positive integers$\}$.

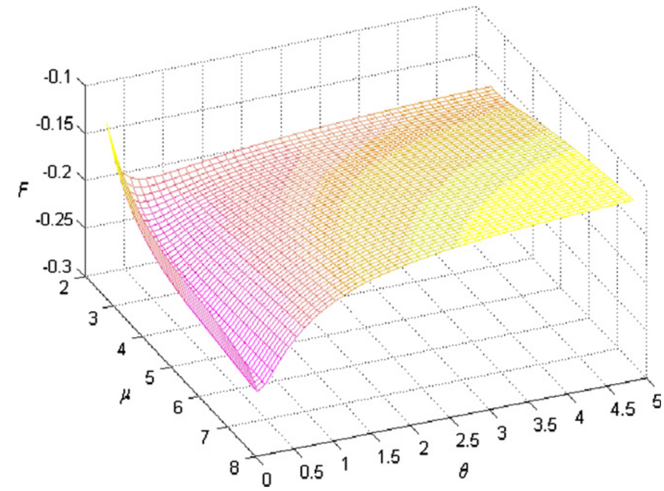

Fig. 3a. Surface plot of cost function.

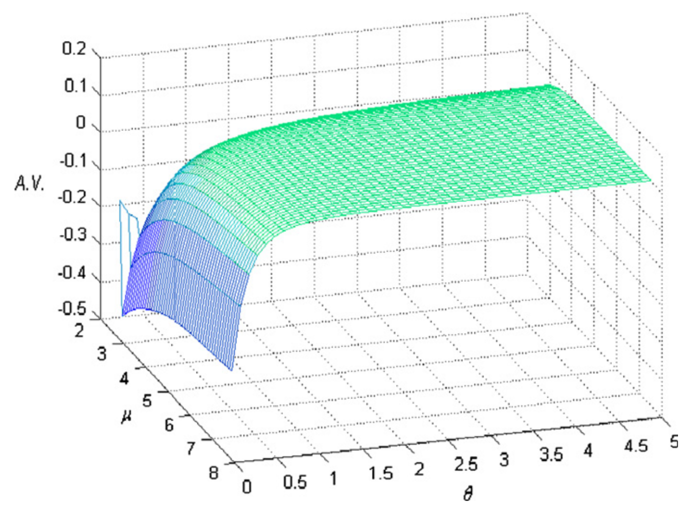

Fig. 3b. Surface plot of $A.V.$

Step 1. Find V^* and K^* for R technicians necessary to satisfy the required availability, where $R = 2, 3, \ldots, 15$ (see Table 1).
Step 2. From Table 1, Θ = {\$1721.24, \$1594.27, \$1509.27, \$1575.78, \$1495.77, \$1609.53, \$1503.85, and \$1544.55}.

Table 1
The expected cost $T_{\text{cost}}(R, V^*, K^*)$ and the system availability $A.V.$ with a synchronous multiple-vacation policy ($\lambda = 1.5$, $\mu = 5.0$, $\theta = 0.5$, $\alpha = 1.0$).

R	10	11	**12**	13	14	15
(R, V^*, K^*)	(10, 2, 2)	(11, 4, 1)	**(12, 3, 2)**	(13, 6, 1)	(14, 4, 2)	(15, 3, 3)
$T_{cost}(R, V^*, K^*)$	1509.27	1575.78	**1495.77**	1609.53	1503.85	1544.55
$A.V.$	0.90531	0.94765	**0.90957**	0.94787	0.91199	0.92006

CPU time spent: 89.092 s.

Table 2
The expected cost $T_{\text{cost}}(R, V^*, K^*)$ and the system availability $A.V.$ with specific multiple-vacation policy ($\lambda = 1.5$, $\mu = 5.0$, $\theta = 0.5$, $\alpha = 1.0$).

R	8	**9**	10	11	12	13
(R, V, K^*)	(8, 1, 1)	**(9, 1, 3)**	(10, 1, 4)	(11, 1, 5)	(12, 1, 6)	(13, 1, 7)
$T_{cost}(R, V, K^*)$	1724.24	**1658.52**	1705.65	1752.55	1799.27	1845.84
$A.V.$	0.94496	**0.90446**	0.91072	0.91603	0.92062	0.92464

CPU time spent: 66.408 s.

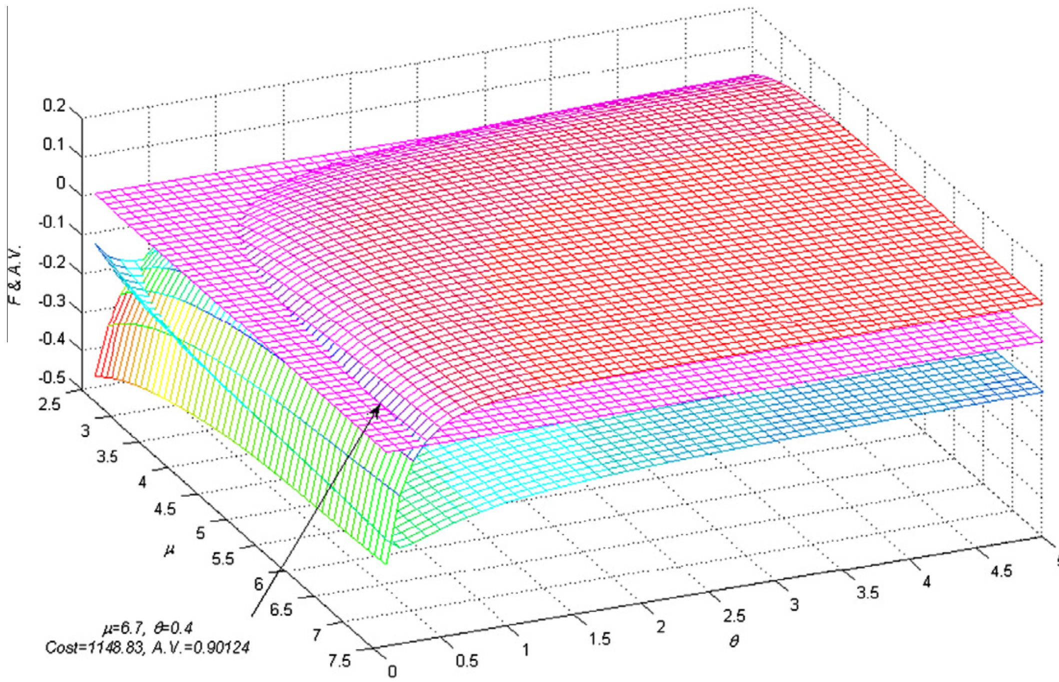

Fig. 4. Cost and availability surfaces for $(M, S, R^*, V^*, K^*) = (15, 10, 12, 3, 2)$ as $\lambda = 1.0$ and $\alpha = 0.5$.

Step 3. From step 2, the optimal solution $T_{\text{cost}}(R^*, V^*, K^*) =$ \$1495.77 is achieved at $R^* = 12$, $K^* = 3$, $V^* = 2$ and the corresponding availability is 0.90957.

To compare the presented machine repair problem with a specific multiple vacation policy, set at the value of team size $V = 1$, and retains restriction K ($K < R$). The corresponding optimal values (R, V, K^*), the expected cost, and the corresponding system availability for this system are as presented in Table 2. It should be noted the steady-state availability does not reach 0.9 for $V = 1$ and $K = R$ cases (traditional multiple vacation policy) from numerical experiments (not presented the manuscript).

As shown in Tables 1 and 2, the synchronous multiple vacation policy has lower (better) minimum cost than the specific multiple vacation policy. As expected, for maintaining the availability level, greater work force allocation flexibility has more advantage. In addition, the specific multiple vacation policy would lead to a shortage in the system availability if allowing all servers on vacation.

In practice, the service rate and vacation rate could be adjusted to further minimize the total cost. It is assumed that the value of service rates and vacation rates have the upper bounds of μ_U and θ_U. The investment budget is restricted by a given budget C. This cost minimization problem could be illustrated mathematically as

$$\begin{aligned} &\underset{0\leqslant\mu\leqslant\mu_U, 0\leqslant\theta\leqslant\theta_U}{\text{Minimize}} \quad F = T_{\text{cost}}(\mu, \theta)/C - 1 \\ &\text{Subject to } A.V. \geqslant A \end{aligned} \tag{40}$$

The object function could be regarded as an alternative form of the original cost function. After fixing the discrete variables, this study addresses the optimization of the continuous variables. For the example provided above, set $\mu_U = 7.5$, $\theta_U = 5.0$, $C = 1500$, Fig. 3a and b displays the surface plots of objective functions and system availability. Three surfaces $T_{\text{cost}}(\mu, \theta)/C - 1$, $A.V. - A$, and $z = 0(xy$ plane) are represented graphically in Fig. 4. It is noted that the points on surface $A.V. - A$ and above $z = 0$ plane are feasible solutions. The optimal service rate μ^* and the optimal vacation rate θ^* are the points achieving the lowest (minimum) cost in the area (feasible region) of $A.V. - A \geqslant 0$ (availability constraint). From Fig. 4, one sees that the optimal solution is $(\mu^*, \theta^*) = (6.7, 0.4)$, and the corresponding minimum object function is -0.23411 $(T_{\text{cost}}(\mu^*, \theta^*) = (1 - 0.23411) \times 1500 = 1148.83)$.

Although the existence of the optimal solution for continuous variables can not be verified due to the complexity of the objective function, we can observe that the optimal solution exists in the range of feasible solutions from the Fig. 4. From many numerical experiments, as expected, we observe that raising the service rate (μ) and the vacation rate (θ) would bring up the system availability to achieve the availability target. In practice use, if there is no feasible solution in the feasible region (constraint violation), some measures (choices) such as adding the budget or expanding the parameter upper bounds should be implemented to fulfill the availability target. That is, there is no feasible solution in the constraint region when the budget is limited and the availability requirement could not be satisfied. In order to obtain the feasible solution, one should consider the increasing of μ_U and θ_U (see Wu & Pearn (2008)), *i.e.*, expand the boundary region. As soon as the availability constrant holds, feasible solutions exist and the optimal solution also exists.

5.1. Sensitivity analysis

A second numerical investigation is performed to deal with the effects of various values of (λ, α) on optimal value (R^*, V^*, K^*), and some system performance measures. The effects of various values of (λ, α) on (R^*, V^*, K^*) are shown in Table 3, which reports that (i) $T_{\text{cost}}(R^*, V^*, K^*)$, R^*, and V^* increase as λ increases. More servers and a greater work force allocation flexibility are required when the machine breakdown rate λ becomes large; (ii) $E[O]$ is maintained at a certain level since the availability requirement; (iii) the increasing of λ or α both leads to a drop of machine availability (*M.A.*).

The third set of numerical results investigates the effects of changing the values of (μ, θ) on optimal value (R^*, V^*, K^*), and some system performance measures. The effects of various values of (μ, θ) on (R^*, V^*, K^*) are shown in Table 4, from which we observe

Table 3
System performance measures of a machine-repair problem with standbys and synchronous multiple vacations under optimal operating conditions ($M = 15$, $S = 10$, $\mu = 5.0$, $\theta = 0.5$).

(λ, α)	(1.0, 1.0)	(1.5, 1.0)	(2.0, 1.0)	(1.0, 0)	(1.0, 0.5)	(1.0, 1.0)
(R^*, V^*, K^*)	(6, 2, 1)	(12, 3, 2)	(13, 4, 1)	(14, 3, 4)	(15, 2, 7)	(6, 2, 1)
T_{cost}	1200.25	1495.77	1770.10	1172.20	1160.75	1200.25
A.V.	0.90906	0.90957	0.90553	0.90653	0.90490	0.90906
$E[F]$	6.07496	6.37355	6.91878	5.30300	5.94204	6.07496
$E[F_q]$	2.28996	1.16842	0.34232	2.35861	2.56221	2.28996
$E[O]$	14.7946	14.7984	14.8011	14.7219	14.7403	14.7946
$E[S]$	4.13039	3.82807	3.28014	4.97506	4.31761	4.13039
$E[B]$	3.78501	5.20513	6.57646	2.94439	3.37983	3.78501
$E[V]$	1.82832	5.82314	3.97739	10.7267	11.4976	1.82832
$E[I]$	0.38667	0.97173	2.44614	0.32892	0.12258	0.38667
M.A	0.75700	0.74506	0.72325	0.78788	0.76232	0.75700
O.U	0.63083	0.43376	0.50588	0.21031	0.22532	0.63083
CPU time	85.484	87.283	87.531	86.314	86.378	87.517

Mean CPU time spent for search (R^*, V^*, K^*): 86.751 s.

Table 4
System performance measures of a machine-repair problem with standbys and synchronous multiple vacations under optimal operating conditions ($\lambda = 1.0$, $\alpha = 0.05$).

(μ, θ)	(2.5, 0.5)	(5.0, 0.5)	(7.5, 0.5)	(5.0, 1.0)	(5.0, 1.5)	(5.0, 2.0)
(R^*, V^*, K^*)	(12, 4, 1)	(7, 2, 2)	(4, 1, 2)	(10, 3, 3)	(7, 3, 2)	(7, 3, 2)
T_{cost}	1679.50	1166.79	1053.38	1038.57	987.908	1004.82
A.V.	0.90227	0.90329	0.91265	0.91822	0.91128	0.94473
$E[F]$	6.52865	5.32927	4.83823	5.19104	5.30587	4.79122
$E[F_q]$	0.54350	2.33186	2.83337	2.19052	2.30984	1.76712
$E[O]$	14.7782	14.7405	14.7668	14.7497	14.7320	14.8527
$E[S]$	3.69314	4.93019	5.39500	5.05928	4.96212	5.35607
$E[B]$	5.98515	2.99741	2.00487	3.00053	2.99603	3.02410
$E[V]$	3.91557	3.61162	1.74900	6.64959	3.67514	3.59499
$E[I]$	2.09928	0.39097	0.24613	0.34989	0.32884	0.38091
M.A.	0.73885	0.78683	0.80647	0.79236	0.78777	0.80835
O.U.	0.49876	0.42820	0.50122	0.30005	0.42800	0.43201
CPU time	88.530	82.983	82.266	85.984	86.391	87.284

Mean CPU time spent for search (R^*, V^*, K^*): 85.573 s.

that (i) $T_{cost}(R^*, V^*, K^*)$, R^*, and V^* decrease as μ increases. Fewer servers are employed and less vacation behaviors are allowed when the server rate becomes large; (ii) the larger μ or θ, the smaller $E[F]$; and (iii) $E[S]$, $E[B]$, *M.A* are insensitive to the change of vacation rate θ.

Similarly, this study is also interested in the effects of (μ^*, θ^*) by changing the values of λ and α. For the fourth set of numerical investigations, this study examines the effects of the various values of λ and α on the optimal values (μ^*, θ^*). Table 5 shows the optimum value (μ^*, θ^*) and several system performance measures for the various values of λ and α, when the optimal values of discrete variables are determined in advance. Table 5 reveals that (i) $E[S]$ is decreased when λ and α are increased, as the increased breakdown rate would raise the substituting behavior and the failure of standby machines; (ii) μ^* is insensitive to the change of λ; and (iii) θ^* increases as λ increases, that is, the vacation time becomes shorter when the arrival rate becomes larger.

6. Conclusions

The systematic methodology provided in this paper works efficiently for a machine repair model with standby under a synchronous multiple vacation policy. The stationary probability vectors were obtained using the matrix-analytical approach and the technique of matrix recursive. First, the steady-state probabilities for this machine repair model were obtained in matrix forms, and then, the explicit expressions for system performance measures, such as the expected number of idle, busy and vacation servers, machine availability, and operative utilization were developed. Next, a cost model was constructed to determine the optimum vacation policy, including the number of technicians R^*, the optimum team size V^*, and the upper bound of the number of vacation teams K^*, in order to reach the minimum cost when system availability is maintained at an acceptable level. Finally, a sensitivity analysis was performed to investigate the affections of joint optimal values (R^*, V^*, K^*) by changing the values of system parameters.

Due to the complexity of the cost formulas, there is no proof of convexity or other properties of the cost function. Moreover, many other algorithms, such as Lagrange relaxation, KKT conditions, steepest descent method, and Quasi-Newton method (Hillier & Lieberman, 2005) could be employed to solve the non-linear optimization problem. A multi-server machine repair problem with synchronous multiple vacation and various teams/groups size, server breakdown, or customer retrial behavior may be good topics for future study.

Table 5
System performance measures of a machine-repair problem with standbys and synchronous multiple vacations under optimal operating conditions ($M = 15$, $S = 10$).

(λ, α)	(1.0, 1.0)	(1.5, 1.0)	(2.0, 1.0)	(1.0, 0)	(1.0, 0.5)	(1.0, 1.0)
(R^*, V^*, K^*)	(6, 2, 1)	(12, 3, 2)	(13, 4, 1)	(14, 3, 4)	(15, 2, 7)	(6, 2, 1)
(μ^*, θ^*)	(5.2, 0.1)	(5.1, 0.2)	(5.1, 0.6)	(7.2, 0.2)	(6.7, 0.4)	(5.2, 0.1)
T_{cost}	1193.25	1492.90	1769.96	1157.39	1148.83	1193.25
A.V.	0.90357	0.90325	0.91517	0.90130	0.90124	0.90357
$E[F]$	6.08886	6.38069	6.78356	4.83871	5.56057	6.08886
$E[F_q]$	2.45210	1.28086	0.30489	2.79536	3.01243	2.45210
$E[O]$	14.7807	14.7797	14.8248	14.7121	14.7056	14.7807
$E[S]$	4.13045	3.83963	3.39169	5.44914	4.73379	4.13045
$E[B]$	3.63676	5.09983	6.47867	2.04335	2.54814	3.63676
$E[V]$	1.96404	5.92432	3.97654	11.6382	12.3396	1.96404
$E[I]$	0.39920	0.97585	2.54479	0.31844	0.11224	0.39920
M.A	0.75645	0.74477	0.72866	0.80645	0.77758	0.75645
O.U	0.60613	0.42499	0.49836	0.14595	0.16988	0.60613
CPU time	399.144	572.044	362.912	1086.61	1620.20	368.077

Mean CPU time spent for search (μ^*, θ^*): 734.831 s.

Acknowledgement

The authors are grateful to the anonymous referees and editor whose constructive comments have led to a substantial improvement in the presentation of the paper.

References

Benson, F., & Cox, D. R. (1951). The productivity of machines requiring attention at random interval. *Journal of the Royal Statistical Society. Series B, 13*, 65–82.
Gupta, S. M. (1997). Machine interference problem with warm spares, server vacations and exhaustive service. *Performance Evaluation, 29*(3), 195–211.
Haque, L., & Armstrong, M. J. (2007). A survey of the machine interference problem. *European Journal of Operational Research, 179*, 469–482.
Hilliard, J. E. (1976). An approach to cost analysis of maintenance float systems. *AIIE Transactions, 8*, 128–133.
Hillier, F. S., & Lieberman, G. J. (2005). *Introduction to operation research* (8th ed.). New York: McGraw-Hill.
Ke, J. C. (2006). Vacation policies for machine interference problem with an unreliable server and state-dependent service rate. *Journal of the Chinese Institute of Industrial Engineers, 23*(2), 100–114.
Ke, J. C., & Lin, C. H. (2005). A markov repairable system involving an imperfect service station with multiple vacations. *Asia–Pacific Journal of Operational Research, 22*(4), 555–582.
Ke, J. C., & Lin, C. H. (2008). Sensitivity analysis of machine repair problems in manufacturing systems with service interruptions. *Applied Mathematical Modelling, 32*, 2087–2105.
Ke, J. C., & Wang, K. H. (1999). Cost analysis of the M/M/R machine repair problem with balking, reneging, and server breakdowns. *Journal of the Operational Research Society, 50*(3), 275–282.
Ke, J. C., & Wang, K. H. (2007). Vacation policies for machine repair problem with two type spares. *Applied Mathematical Modelling, 31*(5), 880–894.

Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models*. Baltimore: Johns Hopkins University Press.
Srinivas, R. C. (2007). A multi-server synchronous vacation model with thresholds and a probabilistic decision rule. *European Journal of Operational Research, 182*, 305–320.
Srinivas, R. C. (2009). Analysis of a multi-server queue with Markovian arrivals and synchronous phase type vacations. *Asia–Pacific Journal of Operational Research, 26*(1), 85–113.
Tian, N., & Zhang, Z. G. (2003). Stationary distributions of GI/M/c queue with PH type vacations. *Queueing Systems, 44*, 183–202.
Tian, N., & Zhang, Z. G. (2006). A two threshold vacation policy in multiserver queueing systems. *European Journal of Operational Research, 168*, 153–163.
Uzsoy, R., Lee, C.-Y., & Martin-Vega, L. A. (1992). A review of production planning and scheduling models in the semiconductor industry. Part I: System characteristics, performance evaluation and production planning. *IIE Transactions, 24*(4), 47–60.
Uzsoy, R., Lee, C.-Y., & Martin-Vega, L. A. (1994). A review of production planning and scheduling models in the semiconductor industry. Part II: Shop-floor control. *IIE Transactions, 26*(5), 44–55.
Wang, K. H., Chen, W. L., & Yang, D. Y. (2009). Optimal management of the machine repair problem with working vacation: Newton's method. *Journal of Computational and Applied Mathematics, 233*, 449–458.
Wang, K. H., & Ke, J. C. (2003). Probabilistic analysis of a repairable system with warm standbys plus balking and reneging. *Applied Mathematical Modelling, 27*, 327–336.
Wang, K. H., Ke, J. B., & Ke, J. C. (2007). Profit analysis of the M/M/R machine repair problem with balking, reneging, and standby switching failures. *Computers and Operations Research, 34*(3), 835–847.
Wu, C. W., & Pearn, W. L. (2008). A variable sampling plan based on C_{pmk} for product acceptance determination. *European Journal of Operational Research, 184*(2), 549–560.
Yue, D., Yue, W., & Sun Y. (2006). Performance analysis of an M/M/c/N queueing system with balking, reneging and synchronous vacations of partial servers. In *ISORA' 06* (pp. 128–143).
Zhang, Z. G., & Tian, N. (2003a). Analysis of queueing systems with synchronous single vacation for some servers. *Queueing Systems, 45*, 161–175.
Zhang, Z. G., & Tian, N. (2003b). Analysis on queueing systems with synchronous vacations of partial servers. *Performance Evaluation, 52*, 269–282.