Computers & Industrial Engineering 131 (2019) 477–487

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Specialized and flexible servers subject to the effects of learning and forgetting

Walid W. Nasr[a,*], Mohamad Y. Jaber[b]

[a] Industrial Engineering and Management, Faculty of Engineering and Architecture, American University of Beirut(AUB), Beirut 1107 2020, Lebanon
[b] Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada

ARTICLE INFO

Keywords:
Queueing
Learning
Forgetting
State-dependent service
Flexible servers
Markovian

ABSTRACT

We consider learning and forgetting in the context of two-server queueing systems and evaluate the tradeoff between utilizing specialized and flexible servers. Specifically, we investigate the performance of two queueing systems. The first system utilizes a specialized workforce where every server handles one job type. The specialized workforce splits the system into two queues where the dedicated servers capitalize on the learning process and consequently reduce the service time. The second system utilizes a flexible workforce where a server can handle any job type. The flexible workforce system allows for the servers to be arranged in parallel where alternating job types results in forgetting and accordingly an increase in service time. A numerical study investigates the impact of the workforce policies on the system performance measures.

1. Introduction

In many industrial settings, managers recognize the presence of learning/forgetting and acknowledge it to be an inherent characteristic of their workforce. Common sense, along with the extensive literature related to workforce learning, indicates that repeating the same job reduces service time and transferring from one job to another can result in longer service times due to forgetting, e.g. Jaber, Kher, and Davis (2003). The incremental fluctuation in performance as a result of learning and forgetting results in state dependent service rates. Depending on the initial learning state of the servers, the incremental change in expected service time obviously increases the time for which the system can be accurately assumed to be in steady state. Delaying the system's convergence to steady state causes the system to function in a transient state for a significant proportion of its operating time. In this work, we consider learning in the context of queueing systems where it is well established that existing approaches to calculate or approximate the behavior of stationary queueing nodes do not accurately capture the transient and time-dependent behavior. Using steady state approximations to describe the behavior of transient systems can lead to inaccurate approximations, Nasr and Taaffe (2012). Consequently, an accurate study of the performance and behavior of queueing nodes exhibiting learning and forgetting should account for the transient behavior of the system. This serves as a motivation for the numerical approach presented in this work which solves for the transient behavior of the queueing systems subject to different workforce policies. The numerical approach is based on defining the states of the Markovian representation of the different queueing systems. This allows us to numerically integrate the Kolmogorov Forward Equations (KFEs) and obtain the transient number-in-system probability distribution for any point in time $t \geqslant 0$.

The effects of learning and forgetting can be observed in many operations in different industries. For instance, the role of learning and forgetting in manufacturing systems and assembly lines has been widely recognized in practice and in the literature, Shafer, Nembhard, and Uzumeri (2001) and Biel and Glock (in press). Recent literature related to learning and forgetting in production and manufacturing systems include Glock and Jaber (2013) and Teyarachakul, mez, and Tarakci (2014) among others. In the context of quality control operations, workers inspecting the quality of products are also subject to learning and forgetting, Giri and Glock (in press). Bollinger and Gillingham (2014) observe significant reductions in cost due to learning in the installation process of solar photovoltaic units in California. The work in Kogan, El Ouardighi, and Herbon (2017) consider learning and forgetting in production systems where a firm's competitive position depends on its accumulated experience. Other real world examples where the impact of an organization's accumulated levels of learning include aircraft production, Benkard (2000). The work in Benkard (2000) emphasizes that forgetting plays an important role when analyzing the dynamics of commercial aircraft production and accounting for learning alone is not sufficient. Other recent applications include Liu, Wang, and Leung (2016) who attribute the variation in the productivity

* Corresponding author.
E-mail address: wn12@aub.edu.lb (W.W. Nasr).

https://doi.org/10.1016/j.cie.2018.02.015
Received 13 September 2017; Received in revised form 3 February 2018; Accepted 10 February 2018
Available online 12 February 2018
0360-8352/

of workstations in cellular manufacturing systems is due to workforce learning and forgetting.

The benefits of an agile and cross-trained workforce is also well established in the literature. The work in Pinker and Shumsky (2000) addresses the trade-off of cross-training workers in the presence of variability in the content of work. Pinker and Shumsky (2000) conclude that cross-trained workers result in a higher throughput, but at the expense of quality. Jordan et al. (2004) consider cross-training workers where the intended applications are maintenance operations at automotive assembly plants. The authors also conclude that achieving a flexible workforce by training workers to perform multiple tasks improves the performance of the system in the presence workload variation. A significant challenge encountered when implementing a flexible workforce is that service time can increase due to variation in the workload. The work in Jaber et al. (2003) captures the increase in service time by accounting for forgetting and relearning when a flexible server is assigned to different workload types. In this work, we also utilize forgetting to capture the behavior of a flexible server. Other applications of flexible workforce include cellular manufacturing systems where its presence is essential to keep up with innovations and new products, Liu et al. (2016). The work in Kaufman, Ahn, and Lewis (2005) investigates the introduction of an agile and temporary workforce into tandem queueing networks and the authors argue that a flexible workforce is becoming more prevalent in manufacturing systems. The work presents an optimal allocation policy, but does not account for learning and forgetting. Lopez and Nembhard (2017) consider learning and forgetting in the context of labor intensive systems. They accordingly present heuristic to compute near optimal solutions for worker assignment problems. Lum et al. (2016) examine several factors that impact learning in a computer based training program. The factors accounted for include eye movement, individual differences in perception of workload, and attention. We also refer to Faccio, Gamberi, Pilati, and Bortolini (2015) and Bortolini, Faccio, Ferrari, Gamberi, and Pilati (2017) for existing literature related to the interaction between workstations and handled products in the context of assembly and packaging systems.

This work assumes that learning and forgetting is an inherent behavioral characteristic of a workforce, which cannot be eliminated. Consequently, the focus of this work is on how to manage and cope with the effects of learning by investigating different workforce policies to capitalize on learning and to counter the effects of forgetting. Specifically, we consider the implications of utilizing a specialized workforce versus a flexible workforce in the presence of learning and forgetting. Surprisingly and despite its importance, our search shows that there is no literature on queueing systems where the service time is dependent on the learning state of the servers and accordingly compares the performance of flexible and specialized policies. We believe this research would also have implications for researchers and practitioners in service systems that are labor intensive. For example, it would be of particular interest and importance for health care operations managers where it is standard practice for nurses or doctors to be assigned to different units such as outpatient clinics, emergency rooms, etc., to encounter shortages in staff and manage workloads, Kaufman et al. (2005).

The rest of the paper is organized as follows. In Section 2, the states of the systems, determined by the accumulated levels of learning and forgetting are presented along with the calculation of the state-dependent service times. The states of the Markovian representation of the queueing systems with flexible and specialized servers are considered in Section 3. The corresponding KFEs are also presented in Section 3 for both queueing systems. The numerical solution for the KFEs is considered in Section 5 for different system parameters where we define measures to evaluate the performance of the queueing systems. A summary discussion, future recommendations and conclusions are presented in Section 6.

2. Model – two-server model with learning/forgetting

Jobs are categorized by two different types and arrive to a two-server system according to a Poisson process with rate λ. The proportion of jobs of type j is p_j for $j = 1,2$, and the arrival rates of jobs of type 1 and 2 are $\lambda_1 = \lambda p_1$ and $\lambda_2 = \lambda p_2$, respectively. We assume the probability that an arriving job is of type j is independent of the type of the preceding or subsequent jobs. We define a measure of the workload variation for an arrival stream with two job types by $V_w = p(1-p)$ where $p_1 = p$ and $p_2 = 1-p$. The maximum workload variation is obtained for $p = 0.5$, which results in the highest frequency of alternating job types.

Two models are investigated. The first model assumes a flexible workforce where a server can work on the next arriving job regardless of type. We refer to this model by the flexible workforce queueing system (FWQS). The second model designates a server to a specific job type. We refer to this model by the specialized workforce queueing system (SWQS). In this work, we only consider two server systems and two job types. The rationale for restricting this paper to two servers is to exploit the importance of considering learning and forgetting in queuing systems where servers perform heterogeneous tasks. Assuming more servers at this stage will complicate the mathematics and will over shadow the importance of these phenomena and defies the purpose of this paper.

2.1. Learning and forgetting

Associating the completion time of a job with the learning state of the server is a commonly utilized modeling approach with a wide range of real world applications. For example, the authors in Lopez and Nembhard (2017) and Nembhard and Bentefouet (2012) denote a worker's performance on a given task as a function of the cumulative number of previous units on that particular task. The intended real world applications are labor intensive systems with learning and forgetting. We utilize a similar representation of the state of a server to represent the accumulated experience level of the workforce. In the context of labor workforce, the work in Bordoloi and Matsuo (2001) define the term knowledge stock to account for the level of learning of a worker with respect to a manufacturing stage. The model in Bordoloi and Matsuo (2001) assume three knowledge levels with respect to two manufacturing stage. Denoting a server's state at time t by the accumulated experience with respect to job types is investigated in the literature and has real world applications, which can include manufacturing systems, call centers, and other labor intensive service systems.

Let $\ell_{i,j}$ be the learning level of server i with respect to type j for $i,j = 1,2$. The learning level parameter, $\ell_{i,j}$, captures the accumulated experience of server i relative to job j. The lowest learning level of server i relative to job j is 1 and is obtained by setting $\ell_{ij} = 1$. Notice that setting $\ell_{i,j} = 1$, either corresponds to server i completing job j for the first time, or server i lost all its accumulated experience in relation to job j due to forgetting. Similarly, we assume a maximum level of experience a server can accumulate relative to a job type. The highest achievable learning level is assumed to be $\ell_{max} > 1$. Accordingly, the learning level parameter $\ell_{i,j}$ varies within the range $[1,\ell_{max}]$ where values of 1 and ℓ_{max} denote the lowest and highest achievable learning levels, respectively.

The learning state of a server is defined by the learning level the server achieved with respect to all job types. The learning state for server i is denoted by $\mathbf{s}_i = (\ell_{i,1},\ell_{i,2})$ for $i = 1,2$. When server i completes a job of type j, learning is accounted for by adjusting the learning state of the server with respect to jobs of type i as follows, $\ell_{i,j} = \min(\ell_{i,j} + 1,\ell_{max})$, for $i,j = 1,2$. Similarly, forgetting is accounted for by lowering the learning state of the server with respect to all other jobs of type $k \neq j$,$\ell_{i,k} = \max(\ell_{i,k}-1,1)$, for all $k \neq j$. Obviously, the fluctuation between learning states is dependent on the workload

variation, V_w. Next, we define the state-dependent service time and rates as a function of the server learning states.

2.2. State-dependent service rates with learning/forgetting

The service time is dependent on the learning state of the system as well as the type of the arriving job. Let $T_{i,j}(\ell)$ be the expected service time of job j by server i when the learning state of job j relative to server i is ℓ, for $i,j = 1,2$, and $\ell = 1,...,\ell_{max}$. Completing job j by server i for the first time is denoted by $T_{i,j}(1)$. Similarly, the corresponding standard time is denoted by $T_{i,j}(\ell_{max})$. The expected service time of job j by server i is a dependent on the learning state of the server and is expressed as (Wright, 1936),

$$T_{i,j}(\ell) = T_{i,j}(1)\ell^{-b_{i,j}} \quad \text{for} \quad i,j = 1,2, \quad \text{and} \quad \ell = 1,...,\ell_{max}, \tag{1}$$

where $b_{i,j}$ is the learning slope. The learning slope $b_{i,j}$, for job j on server i, is expressed as a function of the learning rate $L_{i,j}$ as follows, $0 \leqslant b_{i,j} = \log(L_{i,j})/\log(2) \leqslant 1$. Note that slower/faster values of $L_{i,j}$ result in lower/higher values of the slope $b_{i,j}$, for $i,j = 1,2$. The service rate performance of a job i processed by server j is dependent on the state of the server and is denoted by $\mu_{i,j}(\ell) = 1/T_{i,j}(\ell)$ for $i,j = 1,2$ and $\ell = 1,...,\ell_{max}$.

As a measure of traffic intensity, let ρ_s be the steady state traffic intensity of the two-server system assuming the servers operate at standard time. The steady state parameter ρ_s represents the system traffic intensity assuming the workforce always operate at the maximum experience level and is not subject to learning and forgetting. The standard traffic intensity is calculated as,

$$\rho_s = \frac{\lambda}{\mu_s}, \quad \text{where} \quad \mu_s = \sum_{i=1}^{2}\sum_{j=1}^{2} p_j\, T_{i,j}(\ell_{max}). \tag{2}$$

3. Markovian representation of multi-server system with learning and forgetting

We present a Markovian representation of the two queueing systems. In Section 3.1, the system state-space of the Flexible Workforce queueing system is presented along with the corresponding KFEs. Similarly in Section 3.2, the Markovian representation along with the corresponding KFEs are presented for the Specialized Workforce queueing system.

Notation for the job arrival process, for $j = 1,2$,

- λ: Arrival rate of jobs
- p_j: Probability a job is of type j
- λ_j: Arrival rate of job of type j,$(\lambda_j = \lambda\, p_j)$

Notation for the service process, for $i,j = 1,2$,

- $\ell_{i,j}$: Learning level of server i for job j,$\ell_{i,j} = 1,...,\ell_{max}$
- $\mathbf{s}_i$: Learning state of server i,$\mathbf{s}_i = (\ell_{i,1},\ell_{i,2})$
- $T_{i,j}(\ell)$: Expected service time of job j at server i and a learning state of ℓ, for $\ell = 1,...,\ell_{max}$
- $\mu_{i,j}(\ell)$: Service rate of job j at server i and a learning state of ℓ,$(\mu_{i,j}(\ell) = T_{i,j}^{-1}(\ell))$
- $L_{i,j}$: Learning rate for job j at server i
- $b_{i,j}$: Learning slope for job j at server i

3.1. Flexible workforce

The FWQS entails two servers in parallel where each server has the flexibility to serve all workload types. The service rates are dependent on the type of job and system state relative to the accumulated levels of experience. The variation in the workload results in a fluctuation in the server learning states where completing a certain job type improves the service level of the recently processed job type while simultaneously regressing the learning state of the other job type. At time $t = 0$, we assume that the system starts with no learning experience. This is obtained by setting the service times to $T_{i,j}(1)$ for $i,j = 1,2$. Note that it is possible to set the system at any learning state at time $t = 0$ by initializing the corresponding KFEs.

The Markovian representation of the two-server queueing system at time $t \geqslant 0$ is obtained by augmenting the learning states of the s servers with the number of jobs in the system. Let $N_f(t)$ be the number in system at time t for the FWQS. The learning state of the servers at time t is denoted by $S(t)$. In the case where the number in the system is less than 2, $N_f(t) < 2$ and $t > 0$, then the state of the system is determined by their available servers and the learning state. Denote the occupancy state of the servers at time t by the vector W_t where $W_t(i) = 1,2$ indicates that the i^{th} server is occupied and 0 otherwise.

$$W_t(i) = \begin{cases} 0 & \text{if the } i\text{th server is not occupied} \\ 1 & \text{if the } i\text{th server is occupied with job of type 1} \\ 2 & \text{if the } i\text{th server is occupied with job of type 2} \end{cases} \tag{3}$$

For $N_f(t) < 2$ and $t > 0$, the state probabilities are expressed as $P_{(\omega),\mathbf{s}_1,\mathbf{s}_2}(t) = \text{Prob}(W_t = \omega, S(t) = (\mathbf{s}_1,\mathbf{s}_2))$. Accordingly when $N_f(t) = 1$ and server one is busy, the system state probabilities are denoted by $P_{(\omega),\mathbf{s}_1,\mathbf{s}_2}(t) = P_{(w_1,0),\mathbf{s}_1,\mathbf{s}_2}(t)$ for $w_1 = 1,2$ and $\mathbf{s}_1(k),\mathbf{s}_2(k) = 1,...,\ell_{max}$. Similarly when $N_f(t) = 1$ and server two is busy, $P_{(\omega),\mathbf{s}_1,\mathbf{s}_2}(t) = P_{(0,w_2),\mathbf{s}_1,\mathbf{s}_2}(t)$ for $w_1 = 1,2$. When the system is empty, $N_f(t) = 0$ and $\omega = (0,0)$, the probability distribution of the system states is denoted by $P_{0,\mathbf{s}_1,\mathbf{s}_2}(t)$.

The KFEs for $i < 2$, and $\mathbf{s}_1(k),\mathbf{s}_2(k) = 1,...,\ell_{max}$ for $k = 1,2$ are as follows.

For $\omega = (w_1,0)$ and $w_1 = 1,2$,

$$\begin{aligned} P'_{(w_1,0),\mathbf{s}_1,\mathbf{s}_2}(t) &= -(\mu_{1,w_1}(\ell_{1,w_1}) + \lambda)P_{(w_1,0),\mathbf{s}_1,\mathbf{s}_2}(t) \\ &+ \mu_{2,1}(\ell_{2,1}-1)P_{2,\mathbf{s}_1,\mathrm{F}^1_{\mathbf{s}_2},(w_1,1)}(t)\mathrm{I}_{(\ell_{2,1}>1)} \\ &+ \mu_{2,2}(\ell_{2,2}-1)P_{2,\mathbf{s}_1,\mathrm{F}^2_{\mathbf{s}_2},(w_1,2)}(t)\mathrm{I}_{(\ell_{2,2}>1)} \\ &+ \tfrac{p_{w_1}}{2}\lambda\, P_{0,\mathbf{s}_1,\mathbf{s}_2}(t), \end{aligned} \tag{4}$$

where $\mathrm{F}^j_{\mathbf{s}_k}(j) = \min(\mathbf{s}_k(j)+1,\ell_{max})$ and $\mathrm{F}^j_{\mathbf{s}_k}(i) = \max(\mathbf{s}_k(i)-1,1)$ for $i \neq j$. The indicator function $\mathrm{I}_{(.)} = 1$ if the relation in (.) is satisfied and 0 otherwise. For $\omega = (0,w_2)$ and $w_2 = 1,2$,

$$\begin{aligned} P'_{(0,w_2),\mathbf{s}_1,\mathbf{s}_2}(t) &= -(\mu_{2,w_2}(\ell_{2,w_2}) + \lambda)P_{(0,w_2),\mathbf{s}_1,\mathbf{s}_2}(t) \\ &+ \mu_{1,1}(\ell_{1,1}-1)P_{2,\mathrm{F}^1_{\mathbf{s}_1},\mathbf{s}_2,(1,w_2)}(t)\mathrm{I}_{(\ell_{1,1}>1)} \\ &+ \mu_{1,2}(\ell_{1,2}-1)P_{2,\mathrm{F}^2_{\mathbf{s}_1},\mathbf{s}_2,(2,w_2)}(t)\mathrm{I}_{(\ell_{1,2}>1)} \\ &+ \tfrac{p_{w_2}}{2}\lambda\, P_{0,\mathbf{s}_1,\mathbf{s}_2}(t). \end{aligned} \tag{5}$$

For $\omega = (0,0)$,

$$\begin{aligned} P'_{0,\mathbf{s}_1,\mathbf{s}_2}(t) &= -\lambda\, P_{0,\mathbf{s}_1,\mathbf{s}_2}(t) \\ &+ \mu_1(\ell_{1,1}-1)P_{(1,0),\mathrm{F}^1_{\mathbf{s}_1},\mathbf{s}_2}(t)\mathrm{I}_{(\ell_{1,1}>1)} \\ &+ \mu_1(\ell_{1,2}-1)P_{(2,0),\mathrm{F}^2_{\mathbf{s}_1},\mathbf{s}_2}(t)\mathrm{I}_{(\ell_{1,2}>1)} \\ &+ \mu_2(\ell_{2,1}-1)P_{(0,1),\mathbf{s}_1,\mathrm{F}^1_{\mathbf{s}_2}}(t)\mathrm{I}_{(\ell_{2,1}>1)} \\ &+ p_2\,\mu_2(\ell_{2,2}-1)P_{(0,2),\mathbf{s}_1,\mathrm{F}^2_{\mathbf{s}_2}}(t)\mathrm{I}_{(\ell_{2,2}>1)}. \end{aligned} \tag{6}$$

For $i \geqslant 2$, the system-state probabilities are expressed as, $P_{i,\mathbf{s}_1,\mathbf{s}_2,\omega}(t) = \text{Prob}(N_f(t) = i, S(t) = (\mathbf{s}_1,\mathbf{s}_2), W(t) = \omega)$. The Kolmogorov Forward Equations (KFEs) for $i \geqslant 2, \mathbf{s}_1(k),\mathbf{s}_2(k) = 1,...,\ell_{max}$ for $k = 1,2$, and $\omega = (w_1,w_2)$ for $w_1,w_2 = 1,2$,

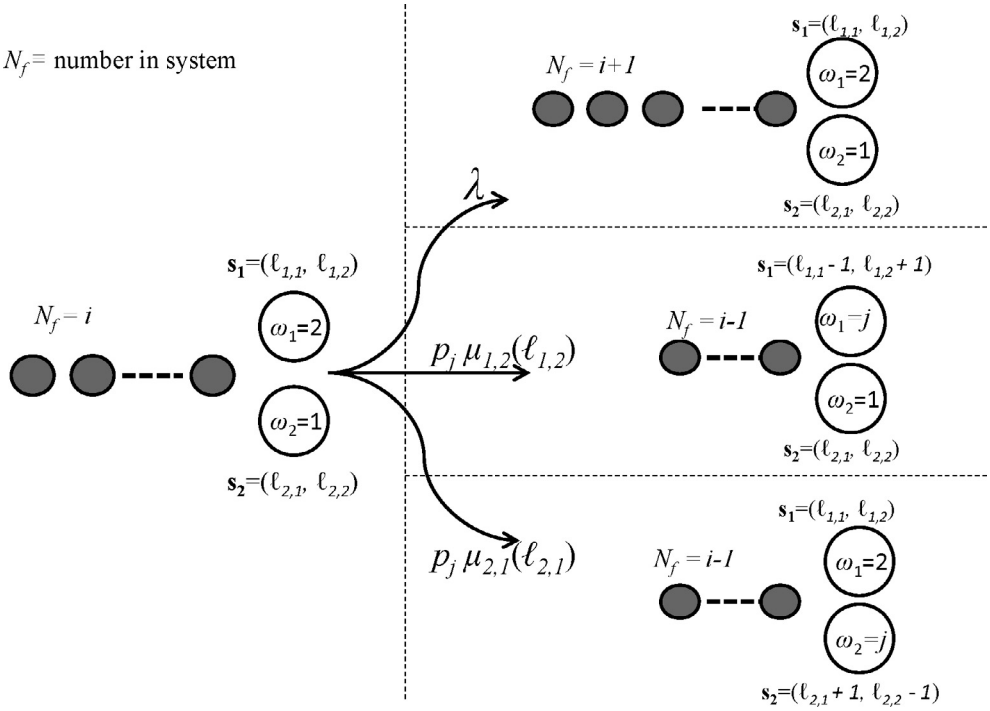

Fig. 1. Markov process transition rates – FWQS example.

$$
\begin{aligned}
&P'_{i,\mathbf{s}_1,\mathbf{s}_2,\boldsymbol{\omega}}(t) = -(\mu_{1,w_1}(\ell_{1,w_1}) + \mu_{2,w_2}(\ell_{2,w_2}) + \lambda\, \mathrm{I}_{(i<c)}) P_{i,\mathbf{s}_1,\mathbf{s}_2,\boldsymbol{\omega}}(t)\\
&+ p_{w_1}\mu_{1,1}(\ell_{1,1}-1) P_{i+1,\mathrm{F}^1_{\mathbf{s}_1},\mathbf{s}_2,(1,w_2)}(t)\mathrm{I}_{(\ell_{1,1}>1)}\\
&+ p_{w_1}\mu_{1,2}(\ell_{1,2}-1) P_{i+1,\mathrm{F}^2_{\mathbf{s}_1},\mathbf{s}_2,(2,w_2)}(t)\mathrm{I}_{(\ell_{1,2}>1)}\\
&+ p_{w_2}\mu_{2,1}(\ell_{2,1}-1) P_{i+1,\mathbf{s}_1,\mathrm{F}^1_{\mathbf{s}_2},(w_1,1)}(t)\mathrm{I}_{(\ell_{2,1}>1)}\\
&+ p_{w_2}\mu_{2,2}(\ell_{2,2}-1) P_{i+1,\mathbf{s}_1,\mathrm{F}^2_{\mathbf{s}_2},(w_1,2)}(t)\mathrm{I}_{(\ell_{2,2}>1)}\\
&+ \lambda\, P_{i-1,\mathbf{s}_1,\mathbf{s}_2,\mathbf{w}}\,\mathrm{I}_{(i\neq 2)}\\
&+ \lambda\,(p_{w_1} P_{(w_1,0),\mathbf{s}_1,\mathbf{s}_2} + p_{w_2} P_{(0,w_2),\mathbf{s}_1,\mathbf{s}_2})\mathrm{I}_{(i=2)}.
\end{aligned}
\tag{7}
$$

As an illustrative example on the behavior of the FWQS Markov process, we consider a system state realization at time $t \geqslant 0$ in Fig. 1. The system state example considered in Fig. 1 has $N_f(t) = i$, server 1 is occupied with a job of type 2, server 2 is occupied with a job of type 1, and the learning state is $S(t) = (\mathbf{s}_1,\mathbf{s}_2) = ((\ell_{1,1},\ell_{1,2}),(\ell_{2,1},\ell_{2,2}))$. An arrival with a rate of λ would result in a transition to state $\{N_f(t) = i+1, S(t) = ((\ell_{1,1},\ell_{1,2}),(\ell_{2,1},\ell_{2,2})), W(t) = (2,1)\}$. A service completion at server 1 results in a transition to state $\{N_f(t) = i-1, S(t) = ((\ell_{1,1}-1,\ell_{1,2}+1),(\ell_{2,1},\ell_{2,2})), W(t) = (j,1)\}$ at a rate of $p_j\mu_{1,2}(\ell_{1,2})$ for $j = 1,2$. Similarly, a service completion at server 2 results in a transition to state $\{N_f(t) = i-1, S(t) = ((\ell_{1,1},\ell_{1,2}), (\ell_{2,1}+1,\ell_{2,2}-1)), W(t) = (2,j)\}$ at a rate of $p_j\mu_{1,2}(\ell_{1,2})$ for $j = 1,2$.

3.2. Specialized workforce

In the SWQS, every job type has a designated server. As a result, two queues can form where the entities within a queue have the same workload content. The system corresponds to two separate queues that share the same system capacity, c. Since every queue handles one type of job, forgetting is not encountered in the SWQS. Every completed job results in learning until the steady state learning level is achieved. The system state space is the resulting Markovian representation is determined by the number of jobs at each queue as well as the learning level at each server. Let $N_1(t)$ and $N_2(t)$ be the numbers in the system at time t for each single server queueing node. The total number in the system at time t for the SWQS becomes $N_s(t) = N_1(t) + N_2(t)$. Although the system is split into two separate queues, the total number in both queues is $\leqslant c-2$. Consequently, the system-states of both queues can not be solved for separately. Accordingly, the system-state probabilities are expressed as, $P_{i,j,s_1,s_2}(t) = \mathrm{Prob}(N_1(t) = i, N_2(t) = j, S(t) = (\mathbf{s}_1,\mathbf{s}_2))$. Notice that in the SWQS case, the server state reduces to $\mathbf{s}_1 = \ell_{1,1}$ and $\mathbf{s}_2 = \ell_{2,2}$. This is a consequence of designating one job type to each server. For simplicity, let $S(t) = (s_1,s_2) = (\ell_{1,1},\ell_{2,2})$ for the SWQS case. The KFEs for $i,j = 0,\ldots,c-1, i+j < c, s_1 = 1,\ldots,\ell_{max}$, and $s_2 = 1,\ldots,\ell_{max}$,

$$
\begin{aligned}
&P'_{i,j,s_1,s_2}(t) = -(\lambda + \mu_1(s_1)\mathrm{I}_{(i>0)} + \mu_2(s_2)\mathrm{I}_{(j>0)})\\
&+ \mu_1(s_1-1) P_{i+1,j,s_1-1,s_2}\,\mathrm{I}_{(s_1>1)}\\
&+ \mu_2(s_2-1) P_{i,j+1,s_1,s_2-2}\,\mathrm{I}_{(s_2>1)}\\
&+ \mu_1(\ell_{max}) P_{i+1,j,\ell_{max},s_2}\,\mathrm{I}_{(s_1=\ell_{max})}\\
&+ \mu_2(\ell_{max}) P_{i,j+1,s_1,\ell_{max}}\,\mathrm{I}_{(s_2=\ell_{max})}\\
&+ \lambda_1 P_{i-1,j,s_1,s_2} + \lambda_2 P_{i,j-1,s_1,s_2}.
\end{aligned}
\tag{8}
$$

The KFEs for $i + j = c, s_1 = 1,\ldots,\ell_{max}$, and $s_2 = 1,\ldots,\ell_{max}$,

$$
P'_{i,j,s_1,s_2}(t) = -(\mu_1(s_1)\mathrm{I}_{(i>0)} + \mu_2(s_2)\mathrm{I}_{(j>0)}) + \lambda_1 P_{i-1,j,s_1,s_2} + \lambda_2 P_{i,j-1,s_1,s_2}.
\tag{9}
$$

As an illustrative example on the behavior of the SWQS Markov process, we consider a system state realization at time $t \geqslant 0$ in Fig. 2. The system state example considered in Fig. 2 at $N_1(t) = i_1, N_2(t) = i_2$, and $S(t) = (s_1,s_2) = (\ell_{1,1},\ell_{2,2})$. An arrival of a job of type 1 with a rate of λ_1 would result in a transition to state $\{N_1(t) = i_1+1, N_2(t) = i_2, S(t) = (\ell_{1,1},\ell_{2,2})\}$. An arrival of a job of type 2 with a rate of λ_2 would result in a transition to state $\{N_1(t) = i_1, N_2(t) = i_2+1, S(t) = (\ell_{1,1},\ell_{2,2})\}$. A service completion at server 1 results in a transition to state $\{N_1(t) = i_1-1, N_2(t) = i_2, S(t) = (\ell_{1,1}+1,\ell_{2,2})\}$ at a rate of $\mu_{1,1}(\ell_{1,1})$. Similarly, a service completion at server 2 results in a transition to state $\{N_1(t) = i_1, N_2(t) = i_2-1, S(t) = (\ell_{1,1},\ell_{2,2}+1)\}$ at a rate of $\mu_{2,2}(\ell_{2,2})$.

Next, we perform an analytical comparison of special cases of the FWQS and SWQS models.

3.3. FWQS and SWQS – analytical comparison

Consider a special case of the FWQS where the service time is always assumed to operate at steady state (i.e., the expected service time is constant and independent of learning and forgetting)

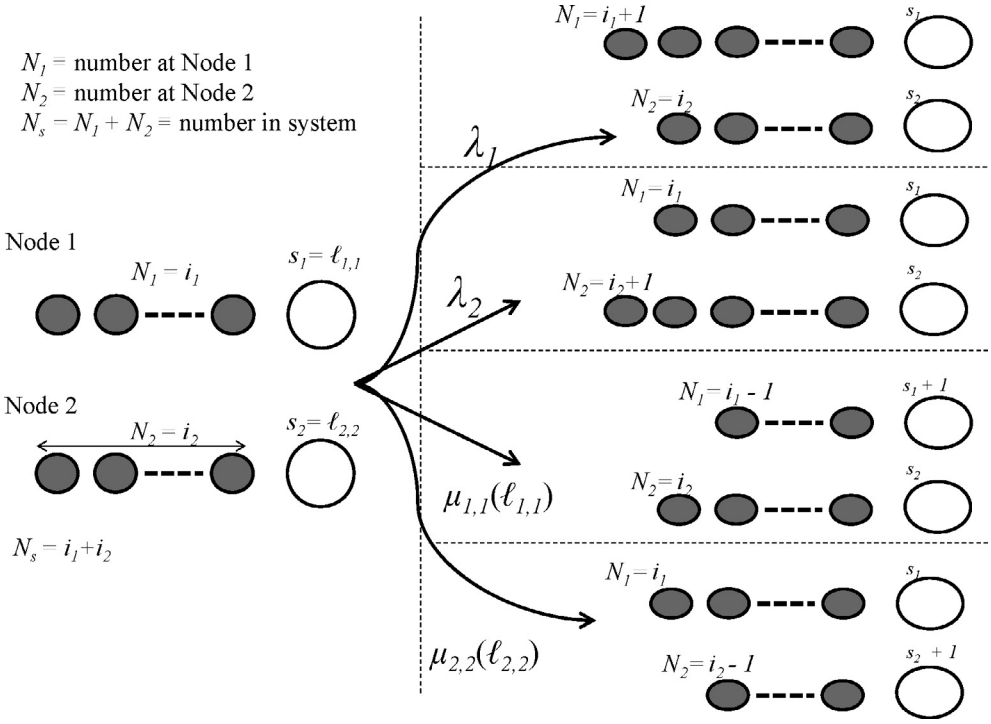

Fig. 2. Markov process transition rates – SWQS example.

and the service time at both servers is identical, $(T_{i,1}(\ell_{max}) = T_{i,2}(\ell_{max}) = T_i(\ell_{max})$ for $i = 1,2)$. In such a case, the service time in the FWQS follows a two state Hyper-Exponential distribution with parameters, $p, \mu_1 = T_1(\ell_{max})^{-1}$ and $\mu_2 = T_2(\ell_{max})^{-1}$, and the FWQS is reduced to an $M/H_2/2$ system. The work in Maddah, Nasr, and Charanek (2017) shows that an $M/H_2/2$ outperforms a specialized two server queueing system except in special cases where the variability of the H_2 distribution is very high. High service time is exhibited in the case where $\mu_1 \gg \mu_2$ and $p_1 \to 0$. The intended applications of this work do not assume such extreme variability in the service time. Accordingly and without loss of generality, we can conclude that (at steady state) the FWQS performs better than the SWQS when the expected service time is constant (assumed to be the standard time) and independent of the learning state.

An obvious advantage of the SWQS queueing system over the FWQS queueing system is that forgetting does not impact the service time. As the impact of forgetting on the service rate increases, the performance of the SWQS improves relative to the FWQS. In such a case, the relative performance of the FWQS and SWQS is dependent on the learning and forgetting parameters and possibly other system parameters. This serves as a further motivation for the mathematical model presented in this work, which allows for a numerical comparison to quantify the advantages of each queueing system. In the next section, we present a numerical comparison to investigate and compare the performance of the FWQS and SWQS models under different system parameters.

3.4. Extending the queueing systems to multiple servers and job types

We generalize the FWQS to account for n_s servers and n_j job types. The flexible workforce model assumes that a server can handle any job type. Accordingly, each server achieves a learning state relative to all job types where $\ell_{i,j}$ is defined for $i = 1,...,n_s$ and $j = 1,...,n_j$. The learning state vector for server i is denoted by $\mathbf{s}_i = (\ell_{i,1},...,\ell_{i,nj})$ for $i = 1,...,n_s$. The occupancy status of the i^{th} server as expressed by Eq. (3) for the two server case is generalized to,

$$W_t(i) = \begin{cases} 0 \text{if the } i\text{th server is not occupied} \\ j \text{if the } i\text{th server is occupied with job of type } j \text{ for} j = 1,...,n_j. \end{cases} \quad (10)$$

A system state of the Markovian representation at time $t \geqslant 0$ is represented by the number-in-system $N_f(t) = 0,...,c$, the learning states of the servers $S(t) = (\mathbf{s}_1,...,\mathbf{s}_{n_s})$ and the status of the ith server $W_t(i)$ for $i = 1,...,n_s$. Consequently, the number of system states is $((n_j + 1)\ell_{max})^{n_s}$ for $N_f(t) \leqslant n_s$, and the number of system states is $(c-n_s)(n_j\ell_{max})^{n_s}$. This results in a total number of $((n_j + 1)\ell_{max})^{n_s} + (c-n_s)(n_j\ell_{max})^{n_s}$ states, which also corresponds to the number of differential equations as represented by the KFEs.

Consider a moderately sized system with three servers ($n_s = 3$), three job types ($n_j = 3$), maximum learning level of 5 ($\ell_{max} = 5$), and a capacity of 20 ($c = 20$). The total number of system states is $((3 + 1) \times 5)^3 + (20-3)(3 \times 5)^3 = 65{,}375$. This corresponds to a set of 63,375 homogeneous differential equations as represented by the KFEs. Accordingly, moderately sized problems would result in a large number of systems states and numerically integrating the KFEs becomes computationally extensive and even prohibitive. Approaches to investigate the performance of such systems can resort to approximating the moment approximations. We refer to Clark (1981), Taaffe and Ong (1987) and Nasr (2008) for approaches to approximate Markovian queueing systems with a large number of system states. Other approaches can include resorting to Monte-Carlo simulation to approximate the behavior of the queueing system at discrete points in time. Here we do not further describe the computational approaches to efficiently approximate the performance of the general FWQS. Extensions of this work can include providing computational approaches and approximations to solve for the key characteristics of the queueing system.

4. Quantifying system performance

Real world applications for which specialized and flexible servers are implemented include car manufacturing industry (Jordan, Inman, & Blumenfeld, 2004), manufacturing and service operations (Iravani, Van Oyen, & Sims, 2005) production systems (Hopp, Tekin, & Van Oyen, 2004), maintenance and service operations (Brusco & Johns, 1998),

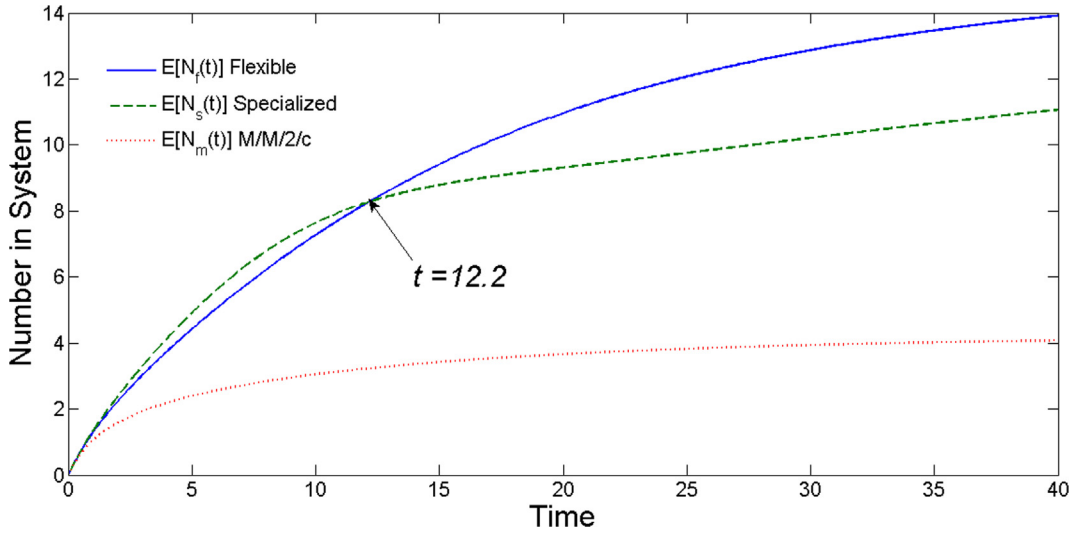

Fig. 3. Number-in-system comparison for FWQS, SWQS and $M/M/2/c$ – ($T_1 = 2, L = 95\%$).

among other real work applications. Cross training workers in the context of manufacturing systems is addressed in Jordan et al. (2004) where the authors quantify the benefits of cross training. Although a flexible workforce can improve the efficiency of a manufacturing process, producing a flexible workforce comes at a cost which can be quantified in terms of training and pooling workers. Pooling workers can lead to a loss of quality, which is emphasized in the context of service operations in Iravani et al. (2005) and Brusco and Johns (1998). This further motivates quantifying the operational performance of flexible and specialized workforces, which results in a more informed decision making process. Accordingly, the savings incurred by utilizing a flexible system should off-set the training and quality costs.

We compare the FWQS and SWQS and also consider the M/M/2/c parallel server system, which assumes the servers always operate at standard time T_s and does not account for learning/forgetting. Let $N_m(t)$ be the number-in-system at time t for the M/M/2/c system operating at standard time. The number-in-system for the M/M/2/c queueing node, $N_m(t)$, is calculated by solving the KFEs for $t \geq 0$,

$$P_0'(t) = -\lambda P_0 + \mu P_1 \text{ and}$$
$$P_i'(t) = -(\lambda \mathrm{I}_{(i<c)} + \max(i,2))P_i(t) + \lambda P_{i-1}(t) + 2\mu P_{i+1}(t)\mathrm{I}_{(i<c)} \text{ for } i = 1,\ldots,c. \quad (11)$$

Performing such a comparison allows a manager to quantify the impact of ignoring learning and forgetting in a parallel server setting. The congestion measures we consider are the number-in-system for the different queueing systems. Accordingly, denote the time-average of the expected number-in-system over the time interval $[0,T]$ by $\overline{\mathrm{E}}[N(t)]$, for the SWQS, FWQS and M/M/2/c systems,

$$\overline{\mathrm{E}}[N_i(T)] = \frac{\int_0^T \mathrm{E}[N_i(t)]dt}{T}, \quad \text{for } i = f,s \text{ and } m,$$

where f,s and m correspond to FWQS, SWQS and the M/M/2/c systems, respectively. Let Δ_1 denote the improvement of utilizing the SWQS over the FWQS,

$$\Delta_1 = \frac{\overline{\mathrm{E}}[N_f(T)] - \overline{\mathrm{E}}[N_s(T)]}{\overline{\mathrm{E}}[N_f(T)]} \times 100.$$

The impact of learning and forgetting on the parallel server system is measured by Δ_2 as follows,

$$\Delta_2 = \frac{\overline{\mathrm{E}}[N_f(T)] - \overline{\mathrm{E}}[N_m(T)]}{\overline{\mathrm{E}}[N_m(T)]} \times 100.$$

Consequently, Δ_2, quantifies the unaccounted cost of ignoring learning/forgetting by assuming that the servers operate at the optimal standard time. High values of Δ_2 indicate that ignoring learning/forgetting

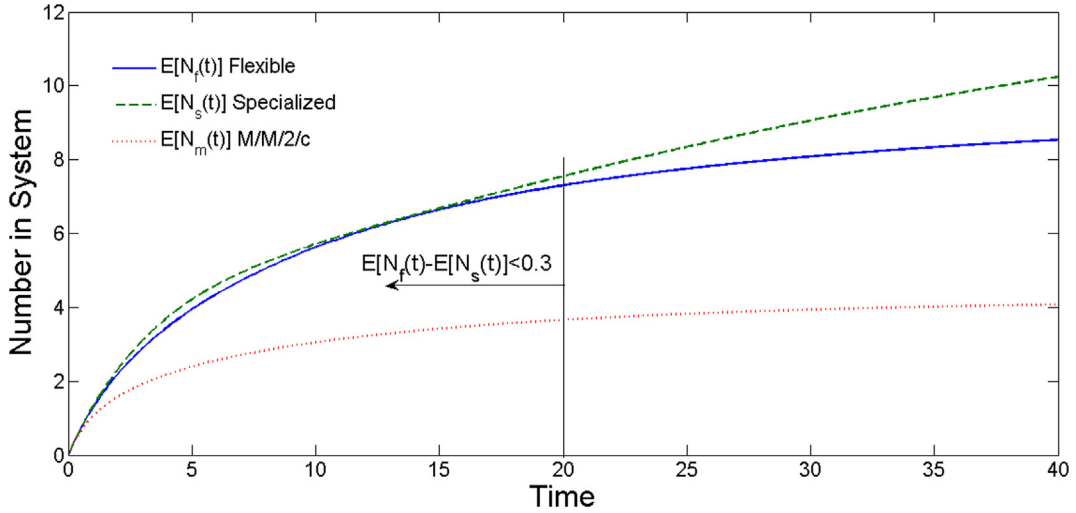

Fig. 4. Number-in-system comparison for FWQS, SWQS and $M/M/2/c$ – ($T_1 = 2, L = 65\%$).

Table 1
Performance of queueing systems – ($T_1 = 2$,$\rho_s = 0.8$). Performance measure: average number-in-system.

	$\bar{E}[N(t)]$				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2
95% (Low)	8.41	9.74	3.35	13.63%	190.83%
85%	7.90	8.66	3.35	8.73%	158.46%
75%	7.44	7.48	3.35	0.60%	123.34%
65% (High)	7.11	6.56	3.35	−8.37%	95.89%

Table 2
Performance of queueing systems – ($T_1 = 2$,$\rho_s = 0.8$). Performance measure: probability of balking.

	β				
L	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	2.74%	6.42%	0.09%	3.68%	6.34%
85%	2.21%	4.14%	0.09%	1.93%	4.06%
75%	1.83%	2.45%	0.09%	0.62%	2.37%
65% (High)	1.60%	1.55%	0.09%	−0.06%	1.46%

Table 3
Cost of queueing systems over time interval $[0,T]$ – ($T_1 = 2$,$\rho_s = 0.8$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
95% (Low)	35.41	43.07	13.45	17.8%	220.2%
85%	33.02	37.28	13.45	11.42%	177.13%
75%	30.91	31.49	13.45	1.84%	134.1%
65% (High)	29.47	27.24	13.45	−8.20%	102.47%

results in misleading performance measures. We further quantify the congestion cost over the time interval $[0,T]$ by $c_n \bar{E}[N_i(T)]T$ for $i = f,s,m$, where c_n denotes the cost per job per unit time.

Another performance measure we consider is the proportion of lost jobs that can not enter the system due to the capacity constraint, which is referred to as balking. Over the time interval $[0,T]$, the number of customers balked is $B(T)$. The expected number of customers balked for each system queueing system is calculated as follows,

$$E[B_i(T)] = \frac{\int_0^T \text{Prob}(N_i(t) = c)dt}{T}\lambda \text{ for } i = f,s \text{ and } m.$$

The percentage of arriving customers that balk over the time interval $[0,T]$ becomes,

$$\beta_i = \frac{E[B_i(T)]}{\lambda T} \times 100 \quad \text{for} \quad i = f,s \text{ and } m.$$

Let c_b denote the cost of not allowing a job to enter the system as a result of balking. Accordingly, the cost of balking over the time interval $[0,T]$ is denoted by $c_b \lambda T(\beta_i/100) = c_b E[B_i(T)]$ for $i = f,s,m$. The total system cost as quantified by the congestion and balking measures over the time interval $[0,T]$ is calculated as $Cost_i(T) = c_n \bar{E}[N_i(T)]T + c_b E[B_i(T)]$ for $i = f,s,m$. From a managerial perspective, a flexible server system is preferred if the cost of cross-training workers and pooling does not exceed $Cost_s(T)$−$Cost_f(T)$.

Table 5
Cost of queueing systems over time interval $[0,T]$ – ($T_1 = 3$,$\rho_s = 0.8$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	46.16	66.29	13.45	30.37%	392.79%
0.85%	42.54	61.61	13.45	30.95%	357.95%
0.75%	39.13	56.24	13.45	30.43%	318.09%
0.65% (High)	36.01	50.26	13.45	28.35%	273.63%

5. Numerical examples

Consider a base case system with two servers and two job types where every server has 5 learning states, $\ell_{max} = 5$. The initial time to complete a job is $T_{i,j}(1) = T_1 = 2$ and the steady state time is half the initial time, i.e., the standard time of a server is $T_{i,j}(\ell_{max}) = T_s = 1$ for $i,j = 1,2$. The service type for $i,j = 1,2$ is assumed as follows,

$$T_{i,j}(\ell)\begin{cases} T_1 & \text{for } \ell = 1, \\ T_1 \ell^{-b_{i,j}} & \text{for } \ell = 2,3,4, \\ T_s & \text{for } \ell = \ell_{max} = 5, \end{cases} \tag{12}$$

where $T_{i,j}(1) = T_1, T_{i,j}(\ell_{max}) = T_s$ and $b_{i,j} = b$, for $i,j = 1,2$. The arrival rate of jobs is $\lambda = 1.6$ where the probability an arriving job is of type 1 or of type 2 is $p_1 = 0.3$ and $p_2 = 1-p_1 = 0.7$, respectively. For the base case example with $\lambda = 1.6$ and $T_s = 1$, the standard utilization of the M/M/2/c system operating at standard time becomes $\rho_s = \frac{\lambda T_s}{2} = 0.8$, Eq. (2).

5.1. Transient analysis

The FWQS and SWQS models are investigated over the time interval $[0,T]$ where the system starts empty at time $t = 0$. The learning state of the servers at time $t = 0$ is initialized to learning states 1, $\ell_{i,j} = 1$ for $i,j = 1,2$. An empty system at $t = 0$ and initializing the learning states to 1 results in a considerable time duration before the system achieve steady state. For this reason, the transient behavior of the system should be closely observed to obtain accurate calculations of the system behavior.

Fig. 3 plots the number of jobs in the system for the FWQS, SWQS and M/M/2/c system using the base case parameters with $L = 95\%$, which results in a slower learning rate. The plots in Fig. 1 illustrate that over the time interval [0,40], a slower learning rate results in a lower average number-in-system for the SWQS. From a managerial perspective, the FWQS performs better if the system runs for short time intervals $t < 12.2$, but on the long runs the SWQS becomes more efficient.

The base case parameters are reconsidered for $L = 65\%$ in Fig. 4

Table 4
Performance measures of queueing systems – ($T_1 = 3$,$\rho_s = 0.8$).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	10.52	13.26	3.35	20.68%	295.78%	6.41%	20.73%	0.09%	14.32%	20.64%
85%	9.85	12.68	3.35	22.30%	278.50%	4.90%	17.02%	0.09%	12.13%	16.94%
75%	9.18	11.93	3.35	23.04%	256.21%	3.75%	13.31%	0.09%	9.56%	13.23%
65%(High)	8.54	10.99	3.35	22.34%	228.20%	2.91%	9.83%	0.09%	6.92%	9.74%

Table 6
Performance measures of queueing systems − ($T_1 = 4, \rho_s = 0.8$).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	12.14	14.59	3.35	16.79%	335.54%	11.98%	31.81%	0.09%	19.83%	31.73%
85%	11.48	14.29	3.35	19.65%	326.57%	9.26%	28.45%	0.09%	19.18%	28.36%
75%	10.76	13.89	3.35	22.49%	314.56%	7.03%	24.75%	0.09%	17.72%	24.66%
65% (High)	10.01	13.35	3.35	24.97%	298.43%	5.30%	20.80%	0.09%	15.50%	20.71%

Table 7
Cost of queueing systems over time interval $[0,T]$ − ($T_1 = 4, \rho_s = 0.8$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	56.22	78.71	13.45	28.57%	485.12%
0.85%	51.85	75.36	13.45	31.2	460.17%
0.75%	47.55	71.38	13.45	33.38%	430.61%
65% (High)	43.45	66.69	13.45	34.85%	395.77%

Table 9
Cost of queueing systems over time interval $[0,T]$ − ($T_1 = 2, \rho_s = 0.7$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	26.38	31.59	9.59	16.49%	229.32%
0.85%	24.29	26.64	9.59	8.80%	177.74%
0.75%	22.51	22.07	9.59	−1.98%	130.14%
0.65% (High)	21.32	18.93	9.59	−12.66%	97.32%

where a faster learning rate resulted in lower number-in-system on average for the FWQS. In Fig. 4, the difference between $N_f(t)$ and $N_s(t)$ does not exceed 0.3 for $t \in [0,20]$. For shorter time intervals, both queueing systems have comparable performances. From a managerial perspective, this gives flexibility to managers as the models are indifferent over the time segment [0,20]. The plots in Figs. 3 and 4 both illustrate that in the case where the servers always operates at standard time, $T_s = 1$, (assuming learning and forgetting do not impact service rates), the performance of the system, as measured by the M/M/2/c system, is superior to both queueing system. This also illustrates that ignoring learning and forgetting by assuming that an expert service is performing at the most efficient service time leads to erroneously optimistic performance measures. Another important observation is the very slow convergence of the FWQS and SWQS compared to the $M/M/2/c$ system. In Figs. 3 and 4, the $M/M/2/c$ system converges to within 3.65% of it steady state value of 4.23 by time $T = 40$. In Fig. 3, the FWQS and SWQS converge to within 10.12% and 20.44% of their steady state values of 15.49 and 13.91, respectively. Similarly in Fig. 4, the FWQS and SWQS converge to within 9.31% and 26.32%, respectively, of their steady state values of 9.42 and 13.90, respectively. This numerically illustrates the slow convergence to steady state and emphasizes the importance of transient analysis.

5.2. Sensitivity analysis

The performance measures of the SWQS and FWQS models are summarized in Tables 1 and 2 for the base case example for different levels of L. For high learning levels, the FWQS results in a smaller number-in-system than the SWQS as illustrated by Δ_1 in Table 1. Not accounting for learning/forgetting results in significant overestimation of the number-in-system as illustrated by Δ_2. We set the number-in-system and balking cost parameters to $c_n = 0.1$\$ per job per unit time and $c_b = 1$\$ per job. Accordingly, Table 3 quantifies the systems costs which decreases for the SWQS and FWQS as the learning rates improve. Table 3 illustrates that improving the learning rate has a larger effect on the FWQS cost which improves from 43.07 to 27.24 (36.75% improvement). The cost of the SWQS is less sensitive to the improved learning rate (relative to the FWQS) and decreases from 35.21 to 29.47 (16.77% improvement).

The balking performance measures are presented in Table 2 for the base case example for different levels of L. The increase in percentage balked when utilizing the FWQS instead of the SWQS is represented by $\delta_1 = \beta_f - \beta_s$ in Table 2. Ignoring learning/forgetting results in an increase of δ_2 of balked jobs where $\delta_2 = \beta_f - \beta_m$. From a managerial perspective, δ_2 quantifies the expected number of balked jobs that are unaccounted for if learning/forgetting is not factored in.

Next, we conduct sensitivity analysis on the system parameters ρ_s, T_1, and V_w. Since a key parameter is the learning level, L, we report the results of the sensitivity analysis relative to the different levels of L. We also note that the range on the learning level considered is 65% to 95%. This is motivated by several studies in the literature that investigate and categorize the different learning rates. Dutton and Thomas (1984) provide a distribution of 108 learning rates collected from different sources and industries. Their results show that the majority of observations are in the neighborhood of 80%. Of all observations, 10 were either very fast (<70%) or very slow (>90%). Dar-El (2013) tabulated learning rates collected from different studies and experiments. The range of the learning rates corroborated the finding of Dutton and

Table 8
Performance measures of queueing systems − ($T_1 = 2, \rho_s = 0.7$).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	6.42	7.44	2.40	13.68%	210.65%	1.07%	2.84%	0.01%	1.77%	2.82%
85%	5.95	6.40	2.40	7.12%	167.30%	0.78%	1.60%	0.01%	0.82%	1.59%
75%	5.53	5.39	2.40	−2.71%	124.81%	0.60%	0.83%	0.01%	0.23%	0.81%
65% (High)	5.25	4.66	2.40	−12.79%	94.34%	0.50%	0.47%	0.01%	−0.02%	0.46%

Table 10
Performance measures of queueing systems $^{-}$ ($T_1 = 2, \rho_s = 0.9$).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	10.37	11.75	4.68	11.74%	151.23%	6.77%	13.58%	0.46%	6.81%	13.12%
85%	9.88	10.79	4.68	8.49%	130.76%	5.83%	9.80%	0.46%	3.97%	9.34%
75%	9.41	9.65	4.68	2.55%	106.39%	5.09%	6.55%	0.46%	1.46%	6.09%
65% (High)	9.07	8.68	4.68	−4.46%	85.59%	4.63%	4.56%	0.46%	−0.07%	4.10%

Table 11
Cost of queueing systems over time interval $[0,T]$ $^{-}$ ($T_1 = 2, \rho_s = 0.9$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	45.81	55.69	19.00	17.74%	193.05%
0.85%	43.23	49.44	19.00	12.55%	160.16%
0.75%	40.88	42.8	19.00	4.48%	125.21%
0.65% (High)	39.23	37.63	19.00	−4.24%	98.04%

Table 13
Cost of queueing systems over time interval $[0,T]$ $^{-}$ ($T_1 = 2, \rho_s = 0.8, V_w = 0.25$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	27.37	48.72	13.45	43.82%	262.16%
0.85%	25.06	42.61	13.45	41.19%	216.73%
0.75%	23.1	36.05	13.45	35.92%	168%
0.65% (High)	21.82	30.53	13.45	28.53%	126.94%

Thomas (1984). By further investigation, Dar-El (2013) classified the learning rates he gathered as pure cognitive (70%), highly cognitive (70−75%), more cognitive than motor (75−80%), more motor than cognitive (80−85%), highly motor (85−90%), and pure motor (90%). Accordingly, a range of 65% to 95% is representative of most industrial operations.

We reconsider the base case for larger values of the starting time T_1 where the service time for the different learning/forgetting states is calculate by Eq. (12). Increasing T_1 results in a larger gap between what an expert can achieve compared to servicing a job for the first time. Tables 4 and 6 present the performance measures (number-in-system and balking) for $T_1 = 3$ and $T_1 = 4$, respectively. Tables 5–7 calculate the systems costs for $T_1 = 3$ and $T_1 = 4$, respectively. Increasing the gap between T_1 and T_s increases the impact of learning (compared to the base case example with $T_1 = 2$) since low learning levels now result in high service times. This further compounds the impact of learning on the performance measures. Consequently, the SWQS provides better results (as illustrated by Δ_1, δ_1 and Δ_{cost_1}). The results are consistent with the observation that as the impact of learning and forgetting increase the performance of the SWQS system improves in comparison with the performance of the FWQS. The impact of ignoring learning/forgetting is very high where Δ_2 ranges between 228.2% and 295.78% when $T_1 = 3$ (Table 4) and ranges between 298.43% and 335.54% when $T_1 = 4$ (Table 6). This further illustrates that ignoring the effects of learning and forgetting on the operational performance of the workforce by assuming steady state service durations can lead to inaccurate and misleading estimates.

Next, we investigate the performance of the SWQS and FWQS models for different traffic intensities. The traffic intensity ρ_s is set to 0.7 ($\lambda = 1.4$) and 0.9 ($\lambda = 1.8$) by varying the arrival rates in Tables 8 and 10, respectively. Tables 9–11 calculate the systems costs for $\rho_s = 0.7$ and $\rho_s = 0.9$, respectively. Comparing Table 8 with the base case Tables 1 and 2, the FWQS performs better for $L = 65\%$ and $L = 75\%$ compared to the base case where the FWQS outperforms the SWQS for $L = 65\%$ only. The results are consistent with the calculations of Tables 10 and 11 where increasing the traffic intensity improves the performance of the SWQS relative to the FWQS. The FWQS performs better for low traffic intensities and high learning rates. This is illustrated by Δ_1 in the last columns of Tables 8, 1 and 10 where for $L = 65\%$ the $\Delta_1 = [-12.79\%, -8.37\%, -4.49\%]$ for $\rho = [0.7, 0.8, 0.9]$, respectively.

We re-consider the base case for different values of V_w which measures the workload variation. We first present in Table 12 the calculations by increasing V_w from 0.21 to 0.25 by setting $p = 0.5$, ($p = 0.3$ in the base case example). As expected, increasing the workload variation results in a larger number-in-system, higher balking rates and higher system costs as can be seen by comparing the entries of Tables 12 and 13 ($V_w = 0.25, p = 0.5$) with the base case Tables 1 and 2 ($p = 0.3$). Positive values of Δ_1 for all learning rates quantify the improvement of utilizing a SWQS over a FWQS for higher workload variability. Also, the significant increase in Δ_2 and Δ_{cost_2} of Tables 12 and 13, compared to the results of the base case, emphasize the impact of ignoring the learning effects in the presence of high variation in the workload. The FWQS consistently outperforms the SWQS for lower workload variability, $p = 0.1$ as illustrated by Δ_1 and Δ_{cost_1} in Tables 14 and 15, respectively. From a managerial perspective, it becomes important to note that the workforce policy should be dependent on the learning level as well as the variation in the workload.

Another important observations is that even for specialized servers,

Table 12
Performance measures of queueing systems $^{-}$ ($T_1 = 2, \rho_s = 0.8, V_w = 0.25$ ($p = 0.5$)).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	6.71	10.71	3.35	37.39%	219.74%	0.86%	9.19%	0.09%	8.33%	9.10%
85%	6.17	9.67	3.35	36.16%	188.64%	0.58%	6.15%	0.09%	5.57%	6.07%
75%	5.71	8.42	3.35	32.20%	151.44%	0.41%	3.69%	0.09%	3.29%	3.61%
65% (High)	5.40	7.28	3.35	25.76%	117.28%	0.32%	2.22%	0.09%	1.89%	2.13%

Table 14
Performance measures of queueing systems $^{-}$ ($T_1 = 2, \rho_s = 0.8, V_w = 0.09$ ($p = 0.1$)).

Average number-in-system						Probability of balking				
	$\bar{E}[N(t)]$					β				
L	SWQS	FWQS	M/M/2/c	Δ_1	Δ_2	SWQS	FWQS	M/M/2/c	δ_1	δ_2
95% (Low)	11.60	6.87	3.35	−68.69%	105.22	11.01%	1.79%	0.09%	−9.21%	1.71%
85%	11.20	6.04	3.35	−85.60%	80.18	9.98%	1.10%	0.09%	−8.88%	1.02%
75%	10.81	5.28	3.35	−104.57%	57.78	9.11%	0.67%	0.09%	−8.45%	0.58%
65% (High)	10.52	4.83	3.35	−117.83%	44.15	8.54%	0.48%	0.09%	−8.06%	0.39%

Table 15
Cost of queueing systems over time interval $[0,T]$ $^{-}$ ($T_1 = 2, \rho_s = 0.8, V_w = 0.09$).

	Cost				
L	SWQS	FWQS	M/M/2/c	Δ_{cost_1}	Δ_{cost_2}
0.95% (Low)	53.43	28.64	13.45	−86.53%	112.92%
0.85%	51.2	24.85	13.45	−106.04%	84.7%
0.75%	49.08	21.57	13.45	−127.56%	60.32%
0.65% (High)	47.53	19.62	13.45	−142.31%	45.83%

convergence time is slow and accordingly the time to achieve the standard service time has to be accounted for. The slow convergence time is further amplified in the FWQS where an increase in V_w increases the probability of alternating the workload and results in slower convergence time. The numerical examples also illustrate that the learning/forgetting rate is not the only system characteristic which needs to be considered when comparing the flexible and specialized queueing systems. The traffic intensity and workload variation also effect the performance of the flexible and specialized workforce queueing systems.

The computational framework presented in this work allows a decision maker to closely investigate the transient behavior of the system at all points in time, which can lead to a more informed decision making process. Figs. 3 and 4 numerically illustrate the applicability of the mathematical model when investigating the transient behavior of the queueing systems by clearly identifying the time range for which a queueing system is favorable. We also present a numerical comparison of the queueing systems over different learning levels that range from 65% to 95%. We note that the range of the learning rate considered, 65–95%, covers the commonly observed learning ranges, Dutton and Thomas (1984), Dar-El (2013). The numerical study observes that ignoring the effects of learning and forgetting by assuming steady state service durations leads to erroneous results and under estimates the costs as quantified by the performance measures. We also illustrate the robustness of the computational framework in terms of conducting sensitivity analysis on the base case example. We observe that increasing the traffic intensity improves the performance of the SWQS relative to the FWQS. Similarly, lower workload variability results in an improved performance of the FWQS relative to the SWQS.

6. Conclusion

This work investigates queueing systems with state-dependent service times and compares the benefits of utilizing flexible and specialized servers. The state of the system at any point in time $t \geqslant 0$, is determined by the learning state of the servers and the number-in-system. Accordingly, for $t \geqslant 0$, every server has a learning state relative to every job type which results in state-dependent service rates. The learning levels fluctuate according to the variation in the content of the workload where learning or forgetting is captured when a job completes service by increasing or decreasing the learning states. We present a numerical approach to capture the performance measures of the FWQS and SWQS. The approached is based on obtaining the state-space of the Markovian representation along with the corresponding KFEs.

An outcome of solving the KFEs is that for any $t \geqslant 0$, the performance of the queueing systems can be closely observed and compared. From a managerial perspective, this results in a more informed and dynamic decision making process. The performance measures include the number-in-system as well as the number of balked customers. Numerical examples are presented for different system parameters which include learning rates, traffic intensity, initial service durations, and workload variation. The examples illustrate that the decision making process is not solely based on the learning rate but should also account for the different system parameters. Another significant managerial observation is that although a certain specialized or flexible workforce policy performs better at steady state, a different policy might perform better for short time periods. Accordingly, the best workforce policy is dependent on the operational time of the system especially if convergence to steady state is slow.

To the best of our knowledge, this is the first attempt to capture the behavior of queueing nodes with state-dependent service times where agile and specialized servers are compared. Like any other piece of scholarship, this work has limitations. One of which is the complexity of the mathematical model, which restricts the numerical examples to two servers with two types of jobs. Otherwise, as explained in an earlier section, the number of system states will be numerous and very difficult to handle. Having a manageable problem size makes it easier for us to interpret the results and for the reader to assimilate the idea and its importance. One may ask why the model was not used to solve an industrial problem. A typical manufacturing facility consists of many and different processes (servers) and jobs. Although the model has been generalized for different numbers of servers and jobs, adopting it in an industrial setting is an endeavor of its own and is outside the scope of this paper. Future work can build on the Markovian representation and consider queueing systems with $s > 2$ servers with $n > 2$ types of job. Increasing the number of servers and job types increases the system-state space, which can significantly increase the computational complexity. In such a case, future work can consider Monte Carlo simulation as an alternative to solving the KFEs.

Acknowledgements

The authors thank the anonymous reviewers and the handling editor, Prof. Yasser Dessouky, for their valuable comments and suggestions. They believe they helped them in improving the presentation of the paper. M.Y. Jaber thanks the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting his research, and the American University of Beirut for the in-kind support.

References

Benkard, C. L. (2000). Learning and forgetting: The dynamics of aircraft production. *American Economic Review, 90*, 1034–1105.

Biel, K., & Glock, C. H. (in press). Governing the dynamics of multi-stage production systems subject to learning and forgetting effects: A simulation study. *International Journal of Production Research* 1–23. < https://doi.org/10.1080/00207543.2017.

1338780 > .
Bollinger, Bryan, K., & Gillingham, Kenneth (2014). *Learning-by-doing in solar photovoltaic installations*. Available at SSRN: < https://ssrn.com/abstract=2342406 or https://doi.org/10.2139/ssrn.2342406 > .
Bordoloi, S. K., & Matsuo, H. (2001). Human resource planning in knowledge-intensive operations: A model for learning with stochastic turnover. *European Journal of Operational Research, 130*(1), 169–189.
Bortolini, M., Faccio, M., Ferrari, E., Gamberi, M., & Pilati, F. (2017). Time and energy optimal unit-load assignment for automatic S/R warehouses. *International Journal of Production Economics, 190*, 133–145.
Brusco, M. J., & Johns, T. R. (1998). Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross training policies. *Decision Sciences, 29*(2), 499–515.
Clark, G. M. (1981). Use of Polya distributions in approximate solutions to nonstationary M/M/s queues. *Communications of the ACM, 24*(4), 206–217.
Dar-El, E. M. (2013). *Human learning: From learning curves to learning organizations*, *Vol. 29*Springer Science & Business Media.
Dutton, J. M., & Thomas, A. (1984). Treating progress functions as a managerial opportunity. *Academy of Management Review, 9*(2), 235–247.
Faccio, M., Gamberi, M., Pilati, F., & Bortolini, M. (2015). Packaging strategy definition for sales kits within an assembly system. *International Journal of Production Research, 53*(11), 3288–3305.
Giri, B. C., & Glock, C. H. (in press). A closed-loop supply chain with stochastic product returns and worker experience under learning and forgetting. *International Journal of Production Research* 1–19. http://dx.doi.org/10.1080/00207543.2017.1347301.
Glock, C. H., & Jaber, M. Y. (2013). A multi-stage production-inventory model with learning and forgetting effects, rework and scrap. *Computers & Industrial Engineering, 64*(2), 708–720.
Hopp, W. J., Tekin, E., & Van Oyen, M. P. (2004). Benefits of skill chaining in serial production lines with cross-trained workers. *Management Science, 50*(1), 83–98.
Iravani, S. M., Van Oyen, M. P., & Sims, K. T. (2005). Structural flexibility: A new perspective on the design of manufacturing and service operations. *Management Science, 51*(2), 151–166.
Jaber, M. Y., Kher, H. V., & Davis, D. J. (2003). Countering forgetting through training and deployment. *International Journal of Production Economics, 85*(1), 33–46.
Jordan, W. C., Inman, R. R., & Blumenfeld, D. E. (2004). Chained cross-training of workers for robust performance. *IIE Transactions, 36*(10), 953–967.
Kaufman, D. L., Ahn, H. S., & Lewis, M. E. (2005). On the introduction of an agile, temporary workforce into a tandem queueing system. *Queueing Systems, 51*(1), 135–171.
Kogan, K., El Ouardighi, F., & Herbon, A. (2017). Production with learning and forgetting in a competitive environment. *International Journal of Production Economics, 189*, 52–62.
Liu, C., Wang, J., & Leung, J. Y. T. (2016). Worker assignment and production planning with learning and forgetting in manufacturing cells by hybrid bacteria foraging algorithm. *Computers & Industrial Engineering, 96*, 162–179.
Lopez, C. E., & Nembhard, D. (2017). Cooperative workforce planning heuristic with worker learning and forgetting and demand constraints. *IIE annual conference. Proceedings* (pp. 380–385). Institute of Industrial and Systems Engineers (IISE).
Lum, H. C., Greatbatch, R. L., Waldfogle, G. E., Benedict, J. D., & Nembhard, D. A. (2016 September). The relationship of eye movement, workload, and attention on learning in a computer-based training program. In: *Proceedings of the human factors and ergonomics society annual meeting*, (Vol. 60(1), pp. 1477–1481). Sage CA (Los Angeles, CA): SAGE Publications. Vancouver.
Maddah, B., Nasr, W. W., & Charanek, A. (2017). A multi-station system for reducing congestion in high-variability queues. *European Journal of Operational Research, 262*(2), 602–619.
Nasr, W. (2008). *Analysis and approximations of time dependent queueing models* (Doctoral dissertation).
Nasr, W. W., & Taaffe, M. R. (2012). Fitting the Pht/Mt/s/c time-dependent departure process for use in tandem queueing networks. *INFORMS Journal on Computing, 25*(4), 758–773.
Nembhard, D. A., & Bentefouet, F. (2012). Parallel system scheduling with general worker learning and forgetting. *International Journal of Production Economics, 139*(2), 533–542.
Pinker, E. J., & Shumsky, R. A. (2000). The efficiency-quality trade-off of cross-trained workers. *Manufacturing & Service Operations Management, 2*(1), 32–48.
Shafer, S. M., Nembhard, D. A., & Uzumeri, M. V. (2001). The effects of worker learning, forgetting, and heterogeneity on assembly line productivity. *Management Science, 47*(12), 1639–1653.
Taaffe, M. R., & Ong, K. L. (1987). Approximating nonstationary Ph(t)/M(t)/s/c queueing systems. *Annals of Operations Research, 8*(1), 103–116.
Teyarachakul, S., mez, D., & Tarakci, H. (2014). Steady-state skill levels of workers in learning and forgetting environments: A dynamical system analysis. *European Journal of Operational Research, 232*(1), 9–21.
Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of Aeronautical Sciences, 3*(4), 122–128.