Probability in the Engineering and Informational Sciences, **22**, 2008, 191–211. Printed in the U.S.A.
doi: 10.1017/S0269964808000119

MAXIMIZING THE THROUGHPUT OF TANDEM LINES WITH FLEXIBLE FAILURE-PRONE SERVERS AND FINITE BUFFERS

Sigrún Andradóttir and Hayriye Ayhan
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
E-mail: sa@isye.gatech.edu; hayhan@isye.gatech.edu

Douglas G. Down
Department of Computing and Software
McMaster University
Hamilton, Ontario L8S 4L7, Canada
E-mail: downd@mcmaster.edu

Consider a tandem queuing network with an infinite supply of jobs in front of the first station, infinite room for completed jobs after the last station, finite buffers between stations, and a number of flexible servers who are subject to failures. We study the dynamic assignment of servers to stations with the goal of maximizing the long-run average throughput. Our main conclusion is that the presence of server failures does not have a major impact on the optimal assignment of servers to stations for the systems we consider. More specifically, we show that when the servers are generalists, any nonidling policy is optimal, irrespective of the reliability of the servers. We also provide theoretical and numerical results for Markovian systems with two stations and two or three servers that suggest that the structure of the optimal server assignment policy does not depend on the reliability of the servers and that ignoring server failures when assigning servers to stations yields near-optimal throughput. Finally, we present numerical results that illustrate that simple server assignment heuristics designed for larger systems with reliable servers also yield good throughput performance in Markovian systems with three stations and three failure-prone servers.

 0269-9648/08 $25.00

1. INTRODUCTION

We study a tandem queuing network with N stations and M flexible servers who are subject to failures. There is an infinite amount of raw material in front of station 1, infinite room for finished jobs after station N, and a finite buffer between stations j and $j+1$, for $j \in \{1, \ldots, N-1\}$, whose size is denoted by B_j. We assume that at any given time, there can be at most one job at each station and that each server can work on at most one job. Moreover, each server $i \in \{1, \ldots, M\}$ works at a deterministic rate $\mu_{ij} \in [0, \infty)$ at each station $j \in \{1, \ldots, N\}$. Thus, server i is trained to work at station j if $\mu_{ij} > 0$. We assume that several workers can work together on a single job, in which case their service rates are additive (i.e., the service mechanism is collaborative). The service requirements of different jobs at station $j \in \{1, \ldots, N\}$ are independent and identically distributed (i.i.d.) random variables with rate $\mu(j)$, which we take to be equal to 1 without loss of generality, and the service requirements at different stations are independent of each other. The lifetimes and repair times of server $i \in \{1, \ldots, M\}$ are i.i.d. random variables with rates $\alpha_i \geq 0$ and $\beta_i > 0$, respectively, and are independent of service requirements ($\alpha_i = 0$ implies that server i never fails, and the repair rates are assumed to be positive to avoid uninteresting cases). For simplicity, we assume that travel and setup times are negligible. Under these assumptions, our objective is to determine the dynamic server assignment policy that maximizes the long-run average throughput.

There is a significant amount of literature on queues with flexible servers. In the interest of space, we do not provide an overview of the entire literature on this subject but refer the interested reader to Andradóttir, Ayhan, and Down [2, 4] and Hopp and Van Oyen [8] for detailed literature reviews. Similarly, there is much of work in the literature on queues with unreliable servers. One can refer to Doshi [7] and Takagi [11] for a survey of the related literature. However, to the best of our knowledge, there are only three articles on queues with flexible unreliable servers, even though server failures are present in many real-life settings. For example, if the servers are humans, failures would correspond to sicknesses, injuries, breaks, and so forth. On the other hand, if the servers are not humans, failures could refer to the instances such as breakdowns and maintenance. Our results indicate that, in certain cases, the optimal server assignment policy is insensitive to server failures and, hence, that plans for the effective usage of servers that do not take server failures into account can be implemented in these cases without incurring substantial deleterious effects.

We now review the previous research on queuing systems with flexible, failure-prone servers and contrast this work with our results. In particular, Wu, Lewis, and Veatch [14] determined the allocation of flexible servers in a clearing system with dedicated and flexible servers, where the dedicated servers are subject to failures. Wu, Down, and Lewis [13] extended these results to serial lines with external arrivals and two stations under the discounted and average cost criteria and developed heuristics for larger systems. Finally, Andradóttir, Ayhan, and Down [3] considered the dynamic assignment of servers to maximize the long-run average throughput of queuing networks with infinite buffers and failure-prone servers and stations. Note that both Wu et al. [14] and Wu et al. [13] assumed that only a subset of the servers are flexible and subject to failures, and both Wu et al. [13] and Andradóttir et al. [3] focused on systems with infinite buffers.

Let Π be the set of all server assignment policies under consideration and let $D_\pi(t)$ denote the number of departures under policy π by time $t \geq 0$. Define

$$T_\pi = \limsup_{t\to\infty} \frac{E[D_\pi(t)]}{t}$$

as the long-run average throughput corresponding to the server assignment policy $\pi \in \Pi$. Our goal is to solve the following optimization problem:

$$\max_{\pi\in\Pi} T_\pi. \quad \textbf{(1)}$$

For two-station tandem lines with $M = 2$ (3) reliable servers and exponentially distributed service times, the optimal server assignment policy was characterized in Andradóttir et al. [2] (Andradóttir and Ayhan [1]). Our results indicate that, for these systems, the structure of the optimal policy remains unchanged when the servers are subject to failures. In particular, when $M = 2$, both servers have primary assignments and leave their primary assignments only to avoid idleness. In other words, we have the somewhat counterintuitive result that the server failures have no effect on the optimal assignment of available servers; (that is, there is no need to compensate for server failures by assigning servers differently to tasks when they are available). For two-station tandem lines with $M = 3$ flexible servers, the optimal policy assigns one of the servers to station 1 unless station 1 is blocked, another server to station 2 unless station 2 is starved, and the third (moving) server to station 1 if the number of the jobs in the buffer is less than a certain value (which could depend on the status of the other servers), and to station 2 otherwise. Thus, the optimal policy is of threshold type both when the servers are always available and also when they might fail. However, the threshold value at which the moving server switches from station 1 to station 2 now also depends on the status of the other servers. For longer tandem lines with generalist servers, Andradóttir et al. [4] showed that any nonidling server assignment policy is optimal. We generalize this result and prove that any nonidling policy (in which servers idle only when they are down) is still optimal when the servers are subject to failures. On the other hand, for longer lines with arbitrary service rates, Andradóttir et al. [2] developed simple server assignment policies for systems in which the number of stations equals the number of (reliable) servers. Our numerical results indicate that these heuristic policies yield good throughput performance even when servers are subject to failures.

The remainder of the article is organized as follows: In Section 2 we provide the optimal server assignment policy for systems with generalist servers. Section 3 focuses on Markovian lines with two stations and two or three servers. In Section 4 we present simple server assignment heuristics for tandem lines with an equal number of servers and stations and use numerical results for Markovian networks with three stations to illustrate that these simple heuristics, in general, yield good throughput performance. Section 5 concludes the paper.

2. SYSTEMS WITH GENERALIST SERVERS

In this section we consider a tandem queue with $M \geq 1$, $N \geq 1$, and generalist servers, so that $\mu_{ij} = \mu_i \gamma_j$ for all $i = 1, \dots, M$ and $j = 1, \dots, N$. Hence, the service rate of each server at each station can be expressed as the product of two constants: one representing the server's speed at every task and the other representing the intrinsic difficulty of the task at the station. We have the following result.

THEOREM 2.1: *Assume that for each $j = 1, \dots, N$, the service requirements $S_{k,j}$ of job $k \geq 1$ at station j are i.i.d. with mean 1. Moreover, assume that for all $t \geq 0$, if there is a job in service at station j at time t, then the expected remaining service requirement at station j of that job is bounded above by a scalar $1 \leq \bar{S} < \infty$. Finally, assume that service is either nonpreemptive or preemptive-resume. If $\mu_{ij} = \mu_i \gamma_j$ for all $i = 1, \dots, M$ and $j = 1, \dots, N$, the lifetimes of server i form a sequence of i.i.d. random variables with rate $0 \leq \alpha_i < \infty$, the repair times of server i form a sequence of i.i.d. random variables with rate $0 < \beta_i < \infty$ and all service requirements, lifetimes, and repair times are independent of each other, then for all $0 \leq B_1, B_2, \dots, B_{N-1} < \infty$, any nonidling server assignment policy π (in which a server idles only when he is down) is optimal, with long-run average throughput*

$$T_\pi = \frac{\sum_{i=1}^{M} \frac{\mu_i \beta_i}{(\alpha_i + \beta_i)}}{\sum_{j=1}^{N} \left(\frac{1}{\gamma_j}\right)}.$$

PROOF: Let $A_\pi(t)$ be the number of jobs that have entered the system by time t under policy $\pi \in \Pi$. Then

$$A_\pi(t) = Q_\pi(t) + D_\pi(t),$$

where $Q_\pi(t)$ denotes the number of customers in the system at time t under policy $\pi \in \Pi$. Since $Q_\pi(t) \leq \sum_{j=1}^{N-1} B_j + N$ for all $t \geq 0$ and for all $\pi \in \Pi$, we have

$$T_\pi = \limsup_{t \to \infty} \frac{\mathbb{E}[D_\pi(t)]}{t} = \limsup_{t \to \infty} \frac{\mathbb{E}[A_\pi(t)]}{t}. \tag{2}$$

Our model is equivalent to one in which the service requirements of successive jobs at station $j \in \{1, \dots, N\}$ are i.i.d. with mean $1/\gamma_j$ and the service rates depend only on the server (i.e., $\mu_{ij} = \mu_i$ for all $i \in \{1, \dots, M\}$). Let π be a nonidling server assignment policy and define $W_{\pi,p}(t)$ as the total work performed by time t for all servers under the policy π. Then $W_{\pi,p}(t) = \sum_{i=1}^{M} \mu_i U_i(t)$, where $U_i(t)$ denotes the total amount of time that server $i \in \{1, \dots, M\}$ is up in the interval $[0, t]$. Let $S_k =$

$\sum_{j=1}^{N} S_{k,\,j}/\gamma_j$ be the total service requirement (in the system) of job k for all $k \geq 1$. Let $W_\pi(t) = \sum_{k=1}^{A_\pi}(t)S_k$ and let $W_{\pi,r}(t) = W_\pi(t) - W_{\pi,p}(t)$ be the total remaining service requirement (work) at time t for the jobs that entered the system by time t. We have

$$\mathbb{E}[W_{\pi,r}(t)] \leq \left(N + \sum_{j=1}^{N-1} B_j\right)\bar{S}\sum_{j=1}^{N}\frac{1}{\gamma_j},$$

which implies that $\lim_{t\to\infty} \mathbb{E}[W_{\pi,\,r}(t)]/t = 0$ and

$$\lim_{t\to\infty}\frac{\mathbb{E}[W_\pi(t)]}{t} = \lim_{t\to\infty}\frac{\mathbb{E}[W_{\pi,p}(t)]}{t} = \sum_{i=1}^{M}\mu_i \lim_{t\to\infty}\frac{\mathbb{E}[U_i(t)]}{t} = \sum_{i=1}^{M}\frac{\mu_i\beta_i}{\alpha_i+\beta_i}, \tag{3}$$

where the last equality follows from a renewal reward process argument when $\alpha_i > 0$ (see, e.g., Ross [10, Sect. 3.6]). For all $n \geq 0$, let $Z_n = (S_{n,1}, \ldots, S_{n,N})$. Since the event $\{A_\pi(t) = n\}$ is completely determined by the random vectors $Z_1, Z_2, \ldots, Z_{n\text{-}1}$ (and independent of $Z_n, Z_{n+1}, \ldots$), $A_\pi(t)$ is a stopping time for the sequence of random vectors $\{Z_n\}$. Moreover, for all $t \geq 0$, $A_\pi(t) \leq K(t)+1$, where $K(t)$ is the number of jobs departing station 1 by time t if all servers work at station 1 at all times, there is unlimited room for completed jobs after station 1, and all servers are up at all times. Since $\{K(t)\}$ is a nondecreasing process with $\lim_{t\to\infty} \mathbb{E}[K(t)]/t = \sum_{i=1}^{M}\mu_i\gamma_1 < \infty$ (which follows from the elementary renewal theorem), we have $\mathbb{E}[A_\pi(t)] < \infty$ for all $t \geq 0$. Then, from Wald's lemma, we have

$$\mathbb{E}[W_\pi(t)] = \mathbb{E}\left[\sum_{k=1}^{A_\pi(t)} S_k\right] = E[A_\pi(t)]\sum_{j=1}^{N}\frac{1}{\gamma_j}. \tag{4}$$

From (2)–(4), we now have

$$\sum_{i=1}^{M}\frac{\mu_i\beta_i}{\alpha_i+\beta_i} = \lim_{t\to\infty}\frac{\mathbb{E}[W_\pi(t)]}{t} = \lim_{t\to\infty}\frac{\mathbb{E}[A_\pi(t)]}{t}\sum_{j=1}^{N}\frac{1}{\gamma_j} = T_\pi\sum_{j=1}^{N}\frac{1}{\gamma_j},$$

which yields the desired throughput. The optimality of this throughput follows from (2)–(4) and the fact that $W_{\pi,p}(t) \leq \sum_{i=1}^{M}\mu_i U_i(t)$ for all $t \geq 0$ and for all server assignment policies $\pi \in \Pi$. ■

Theorem 2.1 shows that for systems with generalist unreliable servers, any non-idling server assignment policy is optimal. This generalizes the corresponding results for reliable servers provided by Andradóttir et al. [2,4]. Note that the proof of Theorem 2.1 is slightly different and simpler than the proofs of the similar results in Andradóttir et al. [2,4] since we made use of the relationship given in (2). Unfortunately, in general (when the servers are not all generalists), assigning servers to tasks in a way that maximizes the throughput is more complex. This issue is addressed in the next two sections.

3. TWO-STATION MARKOVIAN SYSTEMS WITH TWO OR THREE SERVERS

For the remainder of the article we assume that the service requirements, server lifetimes, and server repair times are all exponentially distributed. In this section we consider systems with $N = 2$ stations. For notational convenience, we set $B = B_1$. For all $\pi \in \Pi$ and $t \geq 0$, let $X_\pi(t) = \{X_{\pi,0}(t), X_{\pi,1}(t), \ldots, X_{\pi,M}(t)\}$, where $X_{\pi,0}(t) = 0$ if there is a job to be processed at station 1, the number of jobs waiting to be processed between stations 1 and 2 is zero, and station 2 is starved at time t; $X_{\pi,0}(t) = i$ for $1 \leq i \leq B+1$ if there are jobs to be processed at both stations 1 and 2 and there are $i-1$ jobs waiting to be processed in the buffer at time t; $X_{\pi,0}(t) = B+2$ if station 1 is blocked, B jobs are waiting to be processed in the buffer, and there is a job to be processed at station 2 at time t; and $X_{\pi,j}(t) \in \{0, 1\}$ for $j = 1, \ldots, M$ denotes the status of server j at time t, where 0 refers to the down state and 1 refers to the up state. Let

$$S = \{(i, l_1, \ldots, l_M) : i = 0, 1, \ldots, B+2, l_j = 0, 1 \text{ for } j = 1, \ldots, M\}$$

denote the state space of $\{X_\pi(t): t \geq 0\}$. From now on, we assume that the class Π of server assignment policies under consideration consists of Markovian stationary policies corresponding to the state space S. Then it is clear that $\{X_\pi(t) : t \geq 0\}$ is a continuous-time Markov chain and that there exists a scalar $q_\pi \leq \sum_{i=1}^{M} (\max\{\alpha_i, \beta_i\} + \max_{1 \leq j \leq 2} \mu_{ij}) < \infty$ such that the transition rates $\{q_\pi(s, s')\}$ of $\{X_\pi(t)\}$ satisfy $\sum_{s' \in S, s' \neq s} q_\pi(s, s') \leq q_\pi$ for all $s \in S$. Hence, $\{X_\pi(t)\}$ is uniformizable. Let $\{Y_\pi(k)\}$ be the corresponding discrete-time Markov chain, so that $\{Y_\pi(k)\}$ has state space S and transition probabilities $p_\pi(s, s') = q_\pi(s, s') / q_\pi$ if $s' \neq s$ and $p_\pi(s, s) = 1 - \sum_{s' \in S, s \neq s} q_\pi(s, s') / q_\pi$ for all $s \in S$. Using the analysis in Andradóttir et al. [2, Sect. 3], one can show that the original optimization problem in (1) can be translated into an equivalent (discrete-time) Markov decision problem. More specifically, for all $(i, l_1, \ldots, l_M) \in S$, let

$$R_\pi(i, l_1, \ldots, l_M) = \begin{cases} q_\pi((i, l_1, \ldots, l_M), (i-1, l_1, \ldots, l_M)) & \text{if } i \in \{1, \ldots, B+2\} \\ 0 & \text{if } i = 0 \end{cases}$$

be the departure rate from state $(i, l_1, \ldots, l_M)$ under policy π. Then the optimization problem (1) has the same solution as the Markov decision problem

$$\max_{\pi \in \Pi} \lim_{K \to \infty} \mathbb{E}\left[\frac{1}{K} \sum_{k=1}^{K} R_\pi(Y_\pi(k-1))\right].$$

In other words, maximizing the steady-state throughput of the original queuing system is equivalent to maximizing the steady-state departure rate for the associated embedded (discrete-time) Markov chain.

In Section 3.1, we characterize the optimal server assignment policy for tandem lines with two stations and two servers, and in Section 3.2 we provide the structure of the optimal policy for tandem lines with two stations and three servers. In particular, Theorem 3.1 states the optimal policy for a Markovian system of two stations with one reliable server, one unreliable server, and $0 \leq B \leq 10$, and Theorem 3.2 describes the optimal policy for a Markovian system of two stations with two unreliable servers, and $B=0$. Note that under a Markovian stationary policy π, $\{X_\pi(t)\}$ not only has a much larger state space than the corresponding continuous-time Markov chain for the same system with reliable servers but also is no longer a birth–death process (unlike for systems with reliable servers). Hence, it is difficult to quantify the expressions required in the proofs of Theorems 3.1 and 3.2 for general B. Using our computational resources, we were able to obtain closed-form expressions for these quantities up to buffer size $B=10$ and $B=0$ in Theorems 3.1 and 3.2, respectively. Moreover, we performed a large number of numerical experiments to verify that the policy described in Theorems 3.1 and 3.2 is optimal for systems with $0 < B < \infty$.

3.1. Systems with Two Servers

In this section, we consider a two-station tandem queue with two servers. First, assume that only one of the servers is subject to failures. We now specify the server assignment policy that maximizes the long-run average throughput in this setting for systems with $0 \leq B \leq 10$. Note that our proof of the optimality of the server assignment policy described in Theorem 3.1 differs from the proof of the corresponding result for reliable servers (see Andradóttir et al. [2]) in that we use a linear program rather than the policy iteration algorithm approach to prove the optimality of the server assignment policy.

THEOREM 3.1: *For a Markovian system of two stations with one reliable server, one unreliable server, and* $0 \leq B \leq 10$, *if* $\mu_{11}\mu_{22} \geq \mu_{21}\mu_{12}$, *then the policy that assigns server* 1 *to station* 1 *and server* 2 *to station* 2 *unless station* 1 *is blocked or station* 2 *is starved and assigns both servers to station* 1 *(station* 2*) when station* 2 *(station* 1*) is starved (blocked) is optimal. Moreover, this is the unique optimal policy in the class of Markovian stationary policies if the inequality is strict.*

The uniqueness of the optimal policy in Theorem 3.1 is subject to the interpretation that when a server is down, assigning him to any one of the stations is equivalent to idling him, assigning a server to a station where there is no work is equivalent to idling him, and when $\mu_{ij}=0$, where $i, j \in \{1, 2\}$, assigning server i to station j is equivalent to idling server i. Note also that the optimal policy does not depend on which server is subject to failures. By relabeling the servers, it is clear that Theorem 3.1 shows that when $\mu_{21}\mu_{12} \geq \mu_{11}\mu_{22}$, then the policy that assigns server 1 to station 2 and server 2 to station 1 unless station 1 is blocked or station 2 is starved and assigns both servers to station 1 (station 2) when station 2 (station 1) is starved (blocked) is optimal. Moreover, this is the unique optimal policy in the class of Markovian stationary policies if the inequality is strict.

PROOF: We only provide the proof when server 1 is subject to failures since the proof of the case when server 2 is unreliable is similar and yields the same optimal policy. First, suppose that $\mu_{1_j} = \mu_{2_j} = 0$ for some $j \in \{1, 2\}$ (i.e., there is at least one

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

station at which no server is capable of working). Then the long-run average throughput is zero under any policy and the policy described in Theorem 3.1 is optimal. On the other hand, if $\mu_{i1} = \mu_{i2} = 0$ for some $i \in \{1, 2\}$ (i.e., server i is not capable of working at any station), then Theorem 2.1 shows that any nonidling policy, including the one defined in Theorem 3.1, is optimal. Thus, we can assume without loss of generality that there exist $j_1, j_2 \in \{1, 2\}, j_1 \neq j_2$, such that $\mu_{1_{j1}} > 0$ and $\mu_{2_{j2}} > 0$.

Since only server 1 is subject to failures, the state space of the Markov chain $\{Y_\pi(k)\}$ reduces to $S = \{(0, 1), (0, 0), (1, 1), (1, 0), \dots, (B + 2, 1), (B + 2, 0)\}$. We use the notation $a_{\sigma_1}\sigma_2$ to define the possible actions, where, for $i = 1, 2$, $\sigma_i \in \{I, 1, 2\}$ is the status of server i, with $\sigma_i = I$ if server i is idling and $\sigma_i = j \in \{1, 2\}$ if server i is assigned to station j. Then the set A_s of allowable actions in state s is given as

$$A_s = \begin{cases} \{a_{II}, a_{I1}, a_{1I}, a_{11}\} & \text{for } s = (0, 1) \\ \{a_{1I}, a_{11}\} & \text{for } s = (0, 0) \\ \{a_{II}, a_{I1}, a_{I2}, a_{1I}, a_{2I}, a_{11}, a_{12}, a_{21}, a_{22}\} & \text{for } s = (i, 1), i = 1, \dots, B+1 \\ \{a_{1I}, a_{11}, a_{12}\} & \text{for } s = (i, 0), i = 1, \dots, B+1 \\ \{a_{II}, a_{I2}, a_{2I}, a_{22}\} & \text{for } s = (B+2, 1) \\ \{a_{2I}, a_{22}\} & \text{for } s = (B+2, 0), \end{cases}$$

where we have taken advantage of the equivalence of actions mentioned following the statement of Theorem 3.1. Since the number of possible states and actions are both finite, the existence of an optimal Markovian stationary deterministic policy follows from Theorem 9.1.8 of Puterman [9].

Under our assumptions on the service rates ($\mu_{11}\mu_{22} \geq \mu_{21}\mu_{12}$ and there exist $j_1, j_2 \in \{1, 2\}, j_1 \neq j_2$, such that $\mu_{1_{j1}} > 0$ and $\mu_{2_{j2}} > 0$), neither μ_{11} nor μ_{22} can be equal to zero. Combining this with the assumption that $\beta_1 > 0$, one can deduce that the policy described in Theorem 3.1 corresponds to an irreducible Markov chain and, consequently, that we have a communicating Markov decision process. Thus, one can use the material in Sections 8.8.2 and 9.5.2 of Puterman [9] to prove the optimality of the policy in Theorem 3.1.

Consider the following linear program (P):

$$\text{maximize} \sum_{s \in S} \sum_{a \in A_s} r(s, a)\, x\,(s, a)$$

s.t.

$$\sum_{a \in A_{s'}} x(s', a) - \sum_{s \in S} \sum_{a \in A_s} p(s' | s, a)\, x\,(s, a) = 0 \quad \text{for all } s' \in S,$$

$$\sum_{s \in S} \sum_{a \in A_s} x(s, a) = 1,$$

$$x(s, a) \geq 0 \text{ for all } s \in S, a \in A_s,$$

where for all $s \in S$ and $a \in A_s$, $r\,(s, a)$ is the immediate reward obtained when action a is chosen in state s and $p\,(s' | s, a)$ is the probability of going to state s' in one step when action

a is chosen in state s (see Eqs. 8.8.2), (8.8.3), and (8.8.4) of Puterman [9]). Then Corollary 8.8.7 in Puterman [9] implies that for every basic feasible solution x, there exists at most a single action $a_s \in A_s$ for each $s \in S$ such that $x(s, a_s) > 0$. Consequently, let x^* be a basic feasible optimal solution to (P) and define the decision rule d_{x^*} (which prescribes a procedure for action selection in each state) as

$$d_{x^*}(s) = \begin{cases} a & \text{if } x^*(s, a) > 0 \text{ for } s \in S_{x^*} \\ \text{arbitrary} & \text{for } s \in S \setminus S_{x^*}, \end{cases}$$

where $S_{x^*} = \{s \in S: \sum_{a \in A_s} x(s, a) > 0\}$. Then Corollary 8.8.8 and the discussion on page 483 of Puterman [9] yields that the stationary policy $(d_{x^*})^\infty$ (corresponding to the decision rule d_{x^*}) is optimal. (Note that when using the unichain linear program in communicating models, one cannot select the actions arbitrarily in the states in $S \setminus S_{x^*}$. However, as we show below, the optimal solution of our linear program yields that $S_{x^*} = S$.)

Define

$$d(s) = \begin{cases} a_{11} & \text{if } s = (0, 1) \text{ or } s = (0, 0) \\ a_{12} & \text{if } s = (i, 1) \text{ or } s = (i, 0) \text{ for } i = 1, \ldots, B+1 \\ a_{22} & \text{if } s = (B+2, 1) \text{ or } s = (B+2, 0). \end{cases}$$

Since the Markov chain $\{Y_\pi(k)\}$ under policy $\pi = d^\infty$ is irreducible and has finite state space, its stationary distribution η_d exists. Set $x((0, 1), a_{11}) = \eta_d((0, 1))$, $x((0, 0), a_{11}) = \eta_d((0, 0))$, $x((i, 1), a_{12}) = \eta_d((i, 1))$, and $x((i, 0), a_{12}) = \eta_d((i, 0))$, for all $i = 1, \ldots, B+1$, $x((B+2, 1), a_{22}) = \eta_d((B+2, 1))$, $x((B+2, 0), a_{22}) = \eta_d((B+2, 0))$, and $x(s, a) = 0$ for all other $s \in S$ and $a \in A_s$. Then $S_x = S$, $d_x = d$, and we know from Corollary 8.8.7.b of Puterman [9] that x is a basic feasible solution of (P). In the interest of space, we do not provide the closed-form expressions for the components of x but note that

$$T_{(d_x)^\infty} = \mu_{22}\left[\sum_{i=1}^{B+1}(x((i, 1), a_{12}) + x((i, 0), a_{12})) + x((B+2, 1), a_{22}) + x((B+2, 0), a_{22})\right] + \mu_{12} x((B+2, 1), a_{22}).$$

To prove the optimality of the policy described in Theorem 3.1, it suffices to show that x is an optimal solution of (P). In order to do this, we verify that condition (3.6) of Bazaraa, Jarvis, and Sherali [6, p. 94] is satisfied for all nonbasic variables. Using the notation in Bazaraa et al. [6], let c_B be the vector of the coefficients of the positive elements of x in the objective function of (P) and $\mathbf{B}$ be the matrix of the coefficients of the positive elements of x in the constraint matrix of (P). We have

$$c_B = [0, 0, \mu_{22}, \ldots, \mu_{22}, \mu_{12} + \mu_{22}, \mu_{22}],$$

where the first component corresponds to the coefficient of $x((0, 1), a_{11})$, the second component corresponds to the coefficient of $x((0, 0), a_{11})$, the $(i+2)$th component

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

corresponds to the coefficient of $x(((i+1)/2, 1), a_{12})$ if i is odd and to the coefficient of $x((i/2, 0), a_{12})$ if i is even, for $i = 1, \ldots, 2(B+1)$, and the last two components correspond to the coefficients of $x((B+2, 1), a_{22})$ and $x((B+2, 0), a_{22})$, respectively, in the objective function of (P). Similarly, we have

$$
\mathbf{B} = \left[\begin{array}{cccc}
(\alpha_1 + \mu_{11} + \mu_{21})/q & -\beta_1/q & -\mu_{22}/q & 0 \\
-\alpha_1/q & (\beta_1 + \mu_{21})/q & 0 & -\mu_{22}/q \\
-(\mu_{11} + \mu_{21})/q & 0 & (\alpha_1 + \mu_{11} + \mu_{22})/q & -\beta_1/q \\
0 & -\mu_{21}/q & -\alpha_1/q & (\beta_1 + \mu_{22})/q \\
0 & 0 & -\mu_{11}/q & 0 \\
0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1
\end{array}\right.
$$

$$
\times \left.\begin{array}{ccc}
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & 0 & 0 \\
\cdots & -(\mu_{12} + \mu_{22})/q & 0 \\
\cdots & 0 & -\mu_{22}/q \\
\cdots & (\alpha_1 + \mu_{12} + \mu_{22})/q & -\beta_1/q \\
\cdots & 1 & 1
\end{array}\right],
$$

where we have ordered the states in the same manner as in c_B and q is the uniformization constant. Note that the equation corresponding to state $(B+2, 0)$ is eliminated in (P) since it is redundant (see also Puterman [9, p. 392]). We need to show that

$$
c_B \mathbf{B}^{-1} v_y - c_y \geq 0 \qquad \textbf{(5)}
$$

for each nonbasic variable y, where v_y is the column of the constraint matrix of (P) corresponding to nonbasic variable y and c_y is the coefficient of nonbasic variable y in the objective function of (P). We have obtained closed-form expressions for the difference in (5) for systems with $0 \leq B \leq 10$. In particular, we have

$$c_B\mathbf{B}^{-1}v_{x((0,1),a_{II})} - c_{x((0,1),a_{II})} = T_{(d_x)^\infty} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((0,1),a_{I1})} - c_{x((0,1),a_{I1})} = \frac{\mu_{11}\xi_1}{\xi} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((0,1),a_{1I})} - c_{x((0,1),a_{1I})} = \frac{\mu_{21}\xi_1}{\xi} \geq 0,$$

where ξ and ξ_1 are strictly positive quantities whose expressions depend on B and are omitted in the interest of space. Note that in the last equation we have the expression equal to zero only when $\mu_{21} = 0$, in which case, a_{11} is equivalent to a_{1I}. Similarly,

$$c_B\mathbf{B}^{-1}v_{x((0,0),a_{1I})} - c_{x((0,0),a_{1I})} = T_{(d_x)^\infty} > 0.$$

Now, consider state $(i, 1)$ for $i = 1, \ldots, B+1$. In what follows, $\xi_j, j = 2, 3, \ldots, 9$, are strictly positive quantities that depend on i and B, but their explicit expressions are omitted to conserve space. We have

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{II})} - c_{x((i,1),a_{II})} = T_{(d_x)^\infty} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{I1})} - c_{x((i,1),a_{I1})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_2 + \mu_{11}\xi_3}{\xi} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{I2})} - c_{x((i,1),a_{I2})} = \frac{\mu_{11}\xi_3}{\xi} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{1I})} - c_{x((i,1),a_{1I})} = \frac{\mu_{22}\xi_4}{\xi} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{2I})} - c_{x((i,1),a_{2I})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_5 + \mu_{22}\xi_4}{\xi} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{11})} - c_{x((i,1),a_{11})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_6}{\xi} \geq 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{21})} - c_{x((i,1),a_{21})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_7}{\xi} \geq 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,1),a_{22})} - c_{x((i,1),a_{22})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_8}{\xi} \geq 0.$$

Note that the last three expressions are equal to zero only when $\mu_{11}\mu_{22} - \mu_{21}\mu_{12} = 0$. For state $(i, 0)$, we have

$$c_B\mathbf{B}^{-1}v_{x((i,0),a_{1I})} - c_{x((i,0),a_{1I})} = T_{(d_x)^\infty} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((i,0),a_{11})} - c_{x((i,0),a_{11})} = \frac{(\mu_{11}\mu_{22} - \mu_{21}\mu_{12})\xi_9}{\xi} \geq 0.$$

Note that the equality in the second expression holds only when $\mu_{11}\mu_{22}-\mu_{21}\mu_{12}=0$. Now, consider state $(B+2, 1)$. We have

$$c_B\mathbf{B}^{-1}v_{x((B+2,1),a_{II})} - c_{x((B+2,1),a_{II})} = T_{(d_x)^\infty} > 0,$$

$$c_B\mathbf{B}^{-1}v_{x((B+2,1),a_{I2})} - c_{x((B+2,1),a_{I2})} = \frac{\mu_{12}\xi_{10}}{\xi} \geq 0,$$

$$c_B\mathbf{B}^{-1}v_{x((B+2,1),a_{2I})} - c_{x((B+2,1),a_{2I})} = \frac{\mu_{22}\xi_{10}}{\xi} > 0,$$

where ξ_{10} (whose expression is omitted in the interest of space) is a function of B and is strictly positive. Note that the equality in the second expression holds only when $\mu_{12}=0$, implying that $a_{I2}=a_{22}$. Finally,

$$c_B\mathbf{B}^{-1}v_{x((B+2,0),a_{2I})} - c_{x((B+2,0),a_{2I})} = T_{(d_x)^\infty} > 0.$$

This shows that when $\mu_{11}\mu_{22}-\mu_{21}\mu_{12}\geq 0$, x is an optimal basic feasible solution of (P) and, hence, the policy described in Theorem 3.1 is optimal. It follows from the discussion in Bazaraa et al. [6, p. 104] and the above expressions that if $\mu_{11}\mu_{22}-\mu_{21}\mu_{12}>0$, then x is the unique optimal solution of (P). Combining this with $S_x=S$, we have the uniqueness of the optimal policy in the class of Markovian stationary deterministic policies. ■

The next theorem states that the policy described in Theorem 3.1 remains optimal for systems with $B=0$ when both servers are subject to failures. The proof of Theorem 3.2 is omitted since it is similar to the proof of Theorem 3.1 (except that we have only characterized the difference in (5) for systems with $B=0$ since for systems with $B\geq 1$ the state space is large and the structure of the matrix $\mathbf{B}$ is more complicated than for systems with only one unreliable server).

THEOREM 3.2: *For a Markovian system of two stations, two unreliable servers, and* $B=0$, *if* $\mu_{11}\mu_{22}\geq\mu_{21}\mu_{12}$, *then the policy that assigns server* 1 *to station* 1 *and server* 2 *to station* 2 *unless station* 1 *is blocked or station* 2 *is starved and assigns both servers to station* 1 *(station* 2*) when station* 2 *(station* 1*) is starved (blocked) is optimal. Moreover, this is the unique optimal policy in the class of Markovian stationary policies if the inequality is strict.*

The uniqueness of the optimal policy in Theorem 3.2 is subject to the interpretations mentioned after the statement of Theorem 3.1. Moreover, by relabelling the servers, we have the optimal policy when $\mu_{21}\mu_{12}\geq\mu_{11}\mu_{22}$.

In order to determine if the policy described in Theorems 3.1 and 3.2 is optimal for two-station Markovian systems with two servers and arbitrary buffer size $0\leq B<\infty$, we performed four sets of numerical experiments. In the first two sets of numerical experiments, we considered systems with two stations with one reliable server, one unreliable server, and buffer of size $B\in\{15, 20\}$ between the two stations (recall

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

that B values between 0 and 10 are covered by Theorem 3.1), whereas in the third and fourth sets of numerical examples, we focused on two-station tandem lines with $M = 2$ unreliable servers and buffer size $B \in \{1, \ldots, 5, 10, 15, 20\}$ (recall that $B = 0$ is covered by Theorem 3.2). In the first (third) set of numerical examples, the service rate μ_{ij} of each server $i \in \{1, 2\}$ at each station $j \in \{1, 2\}$, the failure rate α_1 of server 1 (α_i of each server $i \in \{1, 2\}$), and the repair rate β_1 of server 1 (β_i of each server $i \in \{1, 2\}$) are drawn independently from a uniform distribution with range [0, 100]. In the second (fourth) set of examples, we generated the service rate μ_{ij} of each server $i \in \{1, 2\}$ at each station $j \in \{1, 2\}$ from a uniform distribution with range [0, 100], the failure rate α_1 of server 1 (α_i of each server $i \in \{1, 2\}$) from a uniform distribution with range [0, 1], and the repair rate β_1 of server 1 (β_i of each server $i \in \{1, 2\}$) from a uniform distribution with range [0, 10] (all rates were generated independently of one another). Consequently, the first and third sets of examples are concerned with systems in which the relationship among the service times, lifetimes, and repair times is arbitrary, which we will refer to as *systems with common timescales* for the remainder of the article, and the second and fourth sets of examples focus on systems where these three quantities generally happen on different timescales, which will be referred to as *systems with different timescales* for the rest of the article. In each set of examples, we generated 1,000,000 sets of rates independently yielding 1,000,000 different systems for each buffer size B. We then computed the optimal policy for each system considered in the four sets of numerical experiments using the policy iteration algorithm for communicating Markov chains (described in Puterman [9, pp. 479–480]) with the policy given in Theorems 3.1 and 3.2 as the initial policy. In each case, the policy iteration algorithm terminated after one iteration, which implies that no further improvement on throughput is possible. These extensive numerical results demonstrate that the policy described in Theorems 3.1 and 3.2 appears to be optimal for systems with $0 \leq B < \infty$ (at least with high probability). Since this policy is identical to the optimal server assignment policy for a Markovian system of two stations and two reliable servers (see Andradóttir et al. [2, Thm. 4.1], our results suggest that the optimal server assignment policy is insensitive to server failures.

3.2. Systems with Three Servers

In this section we consider a two-station tandem queue with three unreliable servers. The service times, lifetimes, and repair times of all servers are independent and exponentially distributed random variables. The optimal policy for this system when the servers are reliable is given in Andradóttir and Ayhan [1]. We conjecture that the structure of the optimal policy remains unchanged when the servers are subject to failures.

More specifically, we assume that for all $i \in \{1, 2, 3\}$, either $\mu_{i1} > 0$ or $\mu_{i2} > 0$. (If there exists a server i such that $\mu_{i1} = \mu_{i2} = 0$, then the problem reduces to having two servers, which is discussed in the previous section.) Without loss of generality, we also assume that there exist $i, k \in \{1, 2, 3\}$ such that $\mu_{i1} > 0$ and $\mu_{k2} > 0$.

(Note that if $\mu_{11} = \mu_{21} = \mu_{31} = 0$ or $\mu_{12} = \mu_{22} = \mu_{32} = 0$, then the throughput is zero and any policy is optimal.) For $i = 1, 2, 3$, define

$$\rho_i = \frac{\mu_{i1}}{\mu_{i2}},$$

with the convention that a positive real number divided by zero is equal to ∞. Note that ρ_i can be interpreted as the relative skill of server $i \in \{1, 2, 3\}$ at station 1 (as compared to the skill of the server at station 2). Let d, m, and u be such that $\{d, m, u\} = \{1, 2, 3\}$ and $\rho_d \leq \rho_m \leq \rho_u$. Then we conjecture that for each three-tuple (l_1, l_2, l_3), where $l_i \in \{0, 1\}$ denotes the status (down or up) of server $i \in \{1, 2, 3\}$, there exists s^* (l_1, l_2, l_3) such that an optimal server assignment policy $(\delta^*)^\infty$ is given by

$$\delta^*((i, l_1, l_2, l_3)) = \begin{cases} \text{servers } d,\ m, \text{ and } u \text{ work at station 1} \\ \text{servers } m \text{ and } u \text{ work at station 1, server } d \text{ works at station 2} \\ \text{server } u \text{ works at station 1, servers } d \text{ and } m \text{ work at station 2} \\ \text{servers } d,\ m, \text{ and } u \text{ work at station 2} \end{cases}$$

$$\begin{array}{l} \text{for } i = 0 \\ \text{for } 1 \leq i \leq s^*(l_1, l_2, l_3) - 1 \\ \text{for } s^*(l_1, l_2, l_3) \leq i \leq B + 1 \\ \text{for } i = B + 2. \end{array} \quad (6)$$

Note that the above policy generalizes the one in Andradóttir and Ayhan [1] in the sense that the optimal switch point s^* (l_1, l_2, l_3) for server M (where server m moves from station 1 to station 2) can depend on the status of servers d and u. Clearly, if (l_1, l_2, l_3) is such that $l_m = 0$, then s^* (l_1, l_2, l_3) can be chosen arbitrarily.

Proving the optimality of the threshold policy (6) is difficult because the state space $\boldsymbol{S}$ is large even for systems with small buffer sizes, the structure of the matrix $\mathbf{B}$ defined in the proof of Theorem 3.1 is more complicated than for systems with two servers, and the characterization of the optimal switch point is challenging even for systems with reliable servers; see Andradóttir and Ayhan [1]. Consequently, we performed two sets of numerical experiments aimed at determining the structure of the optimal policy for systems with three unreliable servers. More specifically, the first set of numerical examples is concerned with systems with common timescales, and the second set of numerical examples considers systems with different timescales as described in Section 3.1. In each set of examples, the number of replications (sets of service, failure, and repair rates) was again 1,000,000 for systems with $B \in \{0, 1, \ldots, 5, 10, 15\}$ and 100,000 for systems with $B = 20$. (We performed a smaller number of replications for $B = 20$ because the amount of computer time required to find the optimal policy for systems with $B = 20$ is large.) Using the policy iteration algorithm for communicating Markov chains, we computed the optimal policy for each system considered in the two sets of numerical experiments

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

starting with the optimal policy for systems with reliable servers. In each case, the policy iteration algorithm yielded an optimal policy with the structure described in (6). These extensive numerical results demonstrate that policies of this form appear to be optimal for systems with three unreliable servers.

Using the numerical experiments discussed above, we also studied the loss in optimal throughput if one were to choose s^* (l_1, l_2, l_3) independently of l_1, l_2, l_3 (i.e., the switch point for server m does not depend on the status of the servers). To this end, we compared the throughput of the optimal policy with two other policies that are easier to implement. The first policy $(\delta_1)^\infty$ is given by

$$\delta_1((i, l_1, l_2, l_3)) = \begin{cases} \text{servers } d, m, \text{and } u \text{ work at station 1} & \text{for } i = 0 \\ \text{servers } m \text{ and } u \text{ work at station 1, server } d \text{ works at station 2} & \text{for } 1 \le i \le s^* - 1 \\ \text{server } u \text{ works at station 1, servers } d \text{ and } m \text{ work at station 2} & \text{for } s^* \le i \le B + 1 \\ \text{servers } d, m, \text{and } u \text{ work at station 2} & \text{for } i = B + 2, \end{cases}$$

where s^* is the optimal switch point for the corresponding system with reliable servers. The second policy $(\delta_2)^\infty$ is given by

$$\delta_2((i, l_1, l_2, l_3)) = \begin{cases} \text{servers } d, m, \text{and } u \text{ work at station 1} & \text{for } i = 0 \\ \text{servers } m \text{ and } u \text{ work at station 1, server } d \text{ works at station 2} & \text{for } 1 \le i \le s_c^* - 1 \\ \text{server } u \text{ works at station 1, servers } d \text{ and } m \text{ work at station 2} & \text{for } s_c^* \le i \le B + 1 \\ \text{servers } d, m, \text{and } u \text{ work at station 2} & \text{for } i = B + 2, \end{cases}$$

where s_c^* is the (constant) switch point that yields the best throughput among the threshold-type policies (described in this section) with the switch point chosen independently of the status of servers. Tables 1 and 2 display the 95% confidence intervals for the average throughput values of $(\delta_1)^\infty$, $(\delta_2)^\infty$, and $(\delta^*)^\infty$ as a function of the buffer size B for the two sets of numerical experiments described above.

As expected, Tables 1 and 2 demonstrate that the average throughputs achieved by all three policies increase as the buffer size B increases. Moreover, the average throughput of the $(\delta_1)^\infty$ policy is always within 1.41% of the throughput of the optimal policy, and the difference between the average throughputs of the $(\delta_2)^\infty$ policy and the optimal policy never exceeds 0.08%. This shows that the average performance of policies $(\delta_1)^\infty$ and $(\delta_2)^\infty$ is similar to the average

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

Table 1. Throughput Values for Systems with Two Stations, Three Servers, and Common Timescales

B	$(\delta_1)^\infty$	$(\delta_2)^\infty$	$(\delta^*)^\infty$
0	29.9400 ± 0.0282	30.2667 ± 0.0284	30.2707 ± 0.0285
1	31.0931 ± 0.0293	31.3192 ± 0.0295	31.3220 ± 0.0295
2	31.7210 ± 0.0299	31.9085 ± 0.0301	31.9105 ± 0.0301
3	32.1021 ± 0.0304	32.2775 ± 0.0305	32.2789 ± 0.0305
4	32.3517 ± 0.0307	32.5249 ± 0.0308	32.5259 ± 0.0308
5	32.5230 ± 0.0309	32.6991 ± 0.0310	32.6999 ± 0.0310
10	32.9170 ± 0.0315	33.0999 ± 0.0315	33.1002 ± 0.0316
15	33.0589 ± 0.0317	33.2342 ± 0.0318	33.2344 ± 0.0318
20	33.1231 ± 0.1004	33.2861 ± 0.1007	33.2862 ± 0.1007

performance of the optimal policy for all buffer sizes; in particular, $(\delta_2)^\infty$ yields near-optimal throughput. We conclude that choosing the optimal switch point for server m independently of the status of the servers has minimal impact on the throughput, and both policies $(\delta_1)^\infty$ and $(\delta_2)^\infty$ are likely to yield very good performance in practice. Consequently, these numerical results show that using the optimal policy for systems with reliable servers yields near-optimal throughput in systems with unreliable servers.

4. DYNAMIC SERVER ASSIGNMENT POLICIES FOR LARGER SYSTEMS

This section is concerned with server assignment policies for tandem lines with more than two stations when the number of servers is equal to the number of stations. Unfortunately, even when the servers are reliable, the optimal server assignment policy in these larger systems is complicated and might be difficult to implement. Thus, it is important to identify server assignment heuristics with good throughput

Table 2. Throughput Values for Systems with Two Stations, Three Servers, and Different Timescales

B	$(\delta_1)^\infty$	$(\delta_2)^\infty$	$(\delta^*)^\infty$
0	59.6406 ± 0.0396	60.4447 ± 0.0399	60.4910 ± 0.0399
1	62.3232 ± 0.0414	62.7910 ± 0.0415	62.8405 ± 0.0416
2	63.7140 ± 0.0424	64.0210 ± 0.0424	64.0659 ± 0.0424
3	64.5473 ± 0.0430	64.7654 ± 0.0430	64.8032 ± 0.0430
4	65.0900 ± 0.0434	65.2542 ± 0.0433	65.2891 ± 0.0433
5	65.4667 ± 0.0436	65.6000 ± 0.0437	65.6312 ± 0.0436
10	66.3620 ± 0.0442	66.4489 ± 0.0441	66.4690 ± 0.0441
15	66.7196 ± 0.0443	66.8174 ± 0.0443	66.8329 ± 0.0443
20	66.9358 ± 0.1402	67.0578 ± 0.1401	67.0708 ± 0.1401

performance for larger systems. One possible way of achieving this is to determine the best nonmoving policy (i.e., the policy with the highest throughput among those with stationary servers) and apply one step of the policy iteration algorithm on this policy. This approach has been implemented successfully in different settings (see, e.g., Argon, Ding, Glazebrook, and Ziya [5]). However, determining the best nonmoving policy and implementing one step of the policy iteration algorithm could be arduous for larger systems and will not necessarily yield a policy with good throughput performance. Moreover, the resulting policy will depend on the system parameters and could be difficult to implement. By contrast, our objective is to develop easily implementable and robust server assignment heuristics having good throughput performance for a broad range of system parameters. In particular, using numerical experiments, we will illustrate that server assignment heuristics developed for larger systems with reliable servers are also effective in systems with failure-prone servers.

The results provided in Section 3.1 suggest that the optimal policy for systems with two stations and two unreliable servers is the same as for the corresponding systems with reliable servers. This policy has two parts: a primary assignment of servers to stations and a contingency plan specifying what servers will do when there is no work at their primary assignments. The heuristic server assignment policies developed by Andradóttir et al. [2] for larger systems with reliable servers have the same nature (consisting of a primary assignment and a contingency plan). In particular, based on the optimal policy in Section 3.1, for systems with $M = N \geq 2$, the primary assignment of each server $i \in \{1, \ldots, M\}$ is to station $j_i \in \{1, \ldots, N\}$, where $\{j_1, \ldots, j_M\} = \{1, \ldots, N\}$ (so that there is exactly one server at each station) and $\prod_{i=1}^{M} \mu_{ij_i}$ is maximized. Also, we consider the following three contingency plans:

In the first (local) contingency plan, at any time when station $j \in \{1, \ldots, N-1\}$ is blocked, the server with primary assignment at station j will be working downstream at the nearest station $k > j$ where there is work to be done and where there is room for at least one job in the buffer following station k, and at any given time when station $j \in \{2, \ldots, N\}$ is starved but not blocked, the server assigned to station j will be working upstream at the nearest station $k < j$ where there is work to be done.

In the second (push) contingency plan, all servers who have no work to do at the station they are assigned to will be working at the lowest numbered station $1 \leq k \leq N$ that is not blocked.

In the third (pull) contingency plan, all servers that have no work to do at the station they are assigned to will be working at the highest numbered station $1 \leq k \leq N$ that is not starved.

As in Andradóttir et al. [2], when these three contingency plans are implemented with the primary assignment strategy described above, the resulting heuristics will be referred to as the *local*, *push*, and *pull heuristics*.

Andradóttir et al. [2] compared the local, push, and pull heuristics to several other policies (described below) in tandem lines with reliable servers and concluded that these three heuristics (especially the local one) yield near-optimal throughput. We now evaluate the performance of these heuristics in tandem lines with failure-prone servers. To this end, we will compare these three heuristics with several other server assignment policies, including the *optimal policy* and four benchmark policies: the *nonmoving policy* with server i assigned to station i at all times, for all i; the *nonmoving heuristic* using our criterion described in the previous paragraph for assigning servers to stations; the *best nonmoving heuristic* (which yields the largest throughput among all server assignment policies where each server i is assigned to station j_i at all times with $\{j_1, \ldots, j_M\} = \{1, \ldots, N\}$); and the *teamwork* policy of Van Oyen, Gel, and Hopp [12] (which involves assigning all available servers to a single team that will follow each job from the first to the last station and only starts work on a new job once all work on the previous job has been completed). Moreover, we will also compare the three heuristics with the *best local*, *push*, and *pull heuristics* that use the best primary assignment of servers to stations instead of our heuristic primary assignment criterion. The optimal policy and corresponding steady-state throughput were obtained using the policy iteration algorithm.

To evaluate and compare the performance of the 11 server assignment policies described in the previous paragraph, we considered Markovian systems with 3 unreliable servers and 3 stations with a buffer size of $B_1 \in \{1, 2\}$ between stations 1 and 2 and a buffer size of $B_2 \in \{1, 2\}$ between stations 2 and 3. Since the state space of the corresponding Markov chain grows exponentially as the number of stations and the sizes of the buffers increase, we do not consider larger systems (due to not having the required computational resources). As in Section 3, in the first set of examples we considered systems with common time scales, and in the second set of examples we focused on systems with different timescales. Table 3 (Table 4) shows the 95% confidence intervals for the steady-state throughput obtained by each policy for the first (second) set of

TABLE 3. Throughput Values for Systems with Three Stations, Three Servers, and Common Timescales

Policy	Common Buffer Size = 1	Common Buffer Size = 2	Buffer Sizes = Uniform {1, 2}
Optimal policy	27.9501 ± 0.1308	28.7283 ± 0.1344	28.3693 ± 0.1328
Best local heuristic	25.8591 ± 0.1258	26.9705 ± 0.1305	26.4617 ± 0.1287
Best push heuristic	25.6516 ± 0.1229	26.7180 ± 0.1278	26.2563 ± 0.1269
Best pull heuristic	25.6041 ± 0.1238	26.6316 ± 0.1288	26.1731 ± 0.1264
Local heuristic	25.6574 ± 0.1276	26.8185 ± 0.1157	26.2941 ± 0.1087
Push heuristic	25.5405 ± 0.1239	26.6722 ± 0.1285	26.1616 ± 0.1267
Pull heuristic	25.4592 ± 0.1247	26.5085 ± 0.1296	26.0422 ± 0.1272
Teamwork policy	16.4795 ± 0.1627	16.4795 ± 0.1627	16.5499 ± 0.1640
Best nonmoving heuristic	15.9194 ± 0.1026	17.7443 ± 0.1157	16.8536 ± 0.1087
Nonmoving heuristic	15.7929 ± 0.1049	17.5719 ± 0.1186	16.7077 ± 0.1111
Nonmoving policy	9.5728 ± 0.1258	10.4016 ± 0.1407	10.1033 ± 0.1345

Table 4. Throughput Values for Systems with Three Stations, Three Servers, and Different Timescales

Policy	Common Buffer Size = 1	Common Buffer Size = 2	Buffer Sizes = Uniform {1, 2}
Optimal policy	52.2617 ± 0.2405	53.3506 ± 0.2466	52.8443 ± 0.2433
Best local heuristic	48.6272 ± 0.2286	50.4355 ± 0.2374	49.6156 ± 0.2335
Best push heuristic	48.4361 ± 0.2273	50.1688 ± 0.2367	49.3706 ± 0.2318
Best pull heuristic	48.2278 ± 0.2273	49.8551 ± 0.2358	49.1118 ± 0.2311
Local heuristic	48.3514 ± 0.2313	50.2261 ± 0.2396	49.3743 ± 0.2357
Push heuristic	48.2791 ± 0.2289	50.0442 ± 0.2357	49.2331 ± 0.2331
Pull heuristic	48.0083 ± 0.2290	49.6611 ± 0.2372	48.9102 ± 0.2324
Teamwork policy	37.9073 ± 0.4159	37.9073 ± 0.4159	38.1015 ± 0.4191
Best nonmoving heuristic	30.8097 ± 0.1776	33.3873 ± 0.1992	32.1056 ± 0.1874
Nonmoving heuristic	30.5172 ± 0.1839	33.0125 ± 0.2068	31.7824 ± 0.1939
Nonmoving policy	16.9686 ± 0.2256	18.1793 ± 0.2476	17.7924 ± 0.2390

numerical examples for systems with a common buffer size of 1 (i.e., $B_1 = B_2 = 1$), common buffer size of 2 (i.e., $B_1 = B_2 = 2$), and independent and uniformly distributed buffer sizes on the set $\{1, 2\}$ (i.e., $B_1 \sim$ Uniform $\{1, 2\}$, $B_2 \sim$ Uniform $\{1, 2\}$). In each case, the number of replications (sets of service, failure, and repair rates) was 10,000. (Since the amount of computer time required to find the optimal policy for systems with three stations and three servers is large, in this section we performed a smaller number of replications than in Section 3.)

Tables 3 and 4 illustrate that the local, push, and pull heuristics yield good throughput performance on average (e.g., considering the optimal policy as the baseline, the difference between the average throughputs of the optimal policy and the local heuristic is always less than 8.5%). Moreover, the difference between the steady-state throughputs of the local, push, and pull heuristics and the best local, push, and pull heuristics is very small (e.g., considering the best local heuristic as the baseline, the difference between the average throughputs of the best local heuristic and the local heuristic is always less than 0.8%). Among the three heuristics, the local heuristic always shows the best performance and the pull heuristic always shows the worst performance. The performance of the three heuristics is slightly better in the second set of numerical examples, where the service times, lifetimes, and repair times are on different scales. The three heuristics yield much better throughput performance than the nonmoving and teamwork policies. Moreover, the teamwork policy always shows better average behavior than the three nonmoving policies. For the systems considered in Tables 3 and 4, we also implemented various contingency plans where an idle server will give priority to work at a station if the server assigned to that station is failed. Although the results are omitted for reasons of brevity, these heuristics always performed significantly worse than the local, push, and pull heuristics. In summary, the results presented in this section suggest that in

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

tandem lines where the number of servers is equal to the number of stations, server assignment policies designed for systems with reliable servers also achieve very good throughput performance in systems with failure-prone servers.

One heuristic explanation for the insensitivity of desirable server assignment policies to server failures is that the ratio of the effective service rate of each server i at any station j to the server's effective service rate any other station $k \neq j$ does not depend on the reliability of server i (where the effective service rate of server $i \in \{1, \ldots, M\}$ at station $j \in \{1, \ldots, N\}$ is defined as the product of μ_{ij} and the long-run probability $\beta_i/(\alpha_i+\beta_i)$ that server i is up). In fact, the results for flexible reliable servers that only depend on ratios of service rates (i.e, the optimal assignment of generalist servers, the optimal assignment of servers for systems with $M = N = 2$, and the definition of servers d, m, and u for systems with $M = 3$ and $N = 2$) are completely insensitive to server failures, whereas the state where server m moves from station 1 to station 2 for systems with $M = 3$ and $N = 2$ and the optimal server assignment policy for systems with $N = M = 3$ are not completely specified by these ratios, and server failures do in fact have a greater impact in these cases.

5. CONCLUDING REMARKS

The results in this article suggest that in tandem lines with finite buffers, the optimal assignment of servers to stations (in order to achieve good throughput performance) is relatively insensitive to server failures. More specifically, in a system with $N \geq 1$ stations and $M \geq 1$ generalist servers, any nonidling server assignment policy is throughput-optimal, regardless of whether the servers are reliable or unreliable. Similarly, the optimal policy for systems with two stations and two reliable servers also appears to be optimal when the two servers are unreliable. Moreover, the optimal policy for systems with two stations and three unreliable servers has the same structure as the optimal policy for three reliable servers, and even when the optimal policy for systems with unreliable servers does not coincide with the optimal policy for systems with reliable servers, the loss in throughput associated with ignoring server failures by using the optimal policy for reliable servers appears to be insignificant. Finally, in tandem lines with three stations and three servers, server assignment policies designed for larger systems with reliable servers also yield good throughput performance in systems with failure-prone servers.

Acknowledgments

The research of the first author was supported by the National Science Foundation under grants DMI-0217860 and DMI-0400260. The research of the second author was supported by the National Science Foundation under grant DMI-9984352. The research of the third author was supported by the Natural Sciences and Engineering Research Council of Canada. We thank an anonymous referee for a careful review and helpful suggestions.

References

1. Andradóttir, S. & Ayhan, H. (2005). Throughput maximization for tandem lines with two stations and flexible servers. *Operations Research* 53: 516–531.

https://doi.org/10.1017/S0269964808000119 Published online by Cambridge University Press

2. Andradóttir, S., Ayhan, H., & Down, D.G. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems. *Management Science* 47: 1421–1439.
3. Andradóttir, S., Ayhan, H., & Down, D.G. (2007). Compensating for failures with flexible servers. *Operations Research* 55: 753–768.
4. Andradóttir, S., Ayhan, H., & Down, D.G. (2007). Dynamic assignment of dedicated and flexible servers in tandem lines. *Probability in the Engineering and Informational Sciences* 21: 497–538.
5. Argon, N.T., Ding, L., Glazebrook, K.D., & Ziya, S. (2007). Dynamic routing of customers with general delay costs in a multi-server queueing system. *Probability in the Engineering and Informational Sciences* (to appear).
6. Bazaraa, M.S., Jarvis, J.J., & Sherali, H.D. (1990). *Linear programming and network flows*. New York: Wiley.
7. Doshi, H.T. (1986). Queueing systems with vacations: A survey. *Queueing Systems* 1: 29–66.
8. Hopp, W.J. & van Oyen, M.P. (2004). Agile workforce evaluation: A framework for cross-training and coordination. *IIE Transactions* 36: 919–940.
9. Puterman, M.L. (1994). *Markov decision processes*. New York: Wiley.
10. Ross, S.M. (1996). *Stochastic processes*, 2nd ed. New York: Wiley.
11. Takagi, H. (1991). *Queueing analysis: A foundation of performance evaluation. Vol. 1: Vacation and priority systems*. Amsterdam: North-Holland.
12. Van Oyen, M.P., Gel, E.G.S., & Hopp, W.J. (2001). Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Transactions* 33: 761–777.
13. Wu, C.-H., Down, D.G., & Lewis, M.E. (2005). Heuristics for allocation of reconfigurable resources in a serial line with reliability considerations. Preprint.
14. Wu, C.-H., Lewis, M.E., & Veatch, M. (2006). Dynamic allocation of reconfigurable resources in a two-stage tandem queueing system with reliability considerations. *IEEE Transactions on Automatic Control* 51: 309–314.