Flexible Servers in Understaffed Tandem Lines

H. Eser Kırkızlar
School of Management
State University of New York – Binghamton
Binghamton, NY 13902-6000, U.S.A.

Sigrún Andradóttir • Hayriye Ayhan
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205, U.S.A.

Abstract

We study the dynamic assignment of cross-trained servers to stations in understaffed lines with finite buffers. Our objective is to maximize the production rate. We identify optimal server assignment policies for systems with three stations, two servers, different flexibility structures, and either deterministic service times and arbitrary buffers or exponential service times and small buffers. We use these policies to develop server assignment heuristics for Markovian systems with larger buffer sizes that appear to yield near-optimal throughput. In the deterministic setting, we prove that the best possible production rate with full server flexibility and infinite buffers can be attained with partial flexibility and zero buffers, and we identify the critical skills required to achieve this goal. We then present numerical results showing that these critical skills, employed with an effective server assignment policy, also yield near-optimal throughput in the Markovian setting, even for small buffer sizes. Thus our results suggest that partial flexibility is sufficient for near-optimal performance, and that flexibility structures that are effective for deterministic and infinite-buffered systems are also likely to perform well for finite-buffered stochastic systems.

Keywords: Throughput maximization, finite buffers, partial and full server flexibility, tandem production systems, line balancing.

1 Introduction

We consider a production line with $N > 2$ stations and $M \geq 2$ flexible servers. We assume that the line is understaffed (so that $N > M$). We further assume that there is an infinite supply of jobs in front of the first station, infinite room for completed jobs after the last station, and a storage space of size $0 \leq B_j < \infty$ between stations $j-1$ and j, where $j \in \{2, \ldots, N\}$. Our objective is to maximize the long-run average production rate (throughput) in this system.

Understaffed tandem lines with finite buffer spaces and multiple servers are quite typical in the garment manufacturing industry, assembly plants, and warehouses (see, e.g., Bartholdi and Eisenstein, 1996a,b, and Lim and Yang, 2009). In these settings, labor costs constitute a big proportion of the operating costs (see, e.g., Bartholdi and Hackman, 2008), and hence it is important to use the workforce effectively. A possible strategy for understaffed lines is "task partitioning," which involves grouping the tasks and assigning each server to one group of tasks taking into account each server's capabilities (see, e.g., von Hippel, 1990, for a discussion on the benefits of task partitioning). Server flexibility yields improved performance compared to task-portioning because task-partitioning is a special case of flexible server assignment. Rather than strictly partitioning tasks and assigning servers to them, in this paper we consider cross-training structures ranging from full flexibility to zone-training (where servers are trained in consecutive tasks), and using combinations of dedicated and flexible servers. Partial cross-training strategies are especially important in industries where it is costly or impossible to have fully flexible servers, such as when each task requires extensive training or when the number of tasks is large compared to the number of available servers. We focus on tandem lines with two flexible servers and three stations because understaffed lines with more stations are difficult to analyze. However, we also provide some numerical results about effective cross-training strategies in longer lines.

Queues with cross-trained servers have been the subject of a substantial amount of research. Other researchers have addressed the dynamic server assignment problem when $N \geq M = 1$ (Duenyas et al., 1998, Iravani et al., 1997). However, unlike the earlier work, we analyze understaffed lines in the presence of both finite buffers and multiple servers. We now summarize the literature on the dynamic server assignment problem. Hopp and Van Oyen (2004) provide a more detailed review of research on cross-trained workforce.

The majority of the previous work focuses on determining the dynamic assignment policies for flexible servers that minimize holding costs. Most of the papers that will be cited here study parallel or tandem queues with two stations. In particular, Ahn et al. (2002), Ahn et al. (1999), Duenyas et al. (1998), Farrar (1993), Iravani et al. (1997), Kaufman et al. (2005), Rosberg et al. (1982), Wu et al. (2008), and Wu et al. (2006) all study tandem queues with flexible servers, and show that most of the time the optimal server assignment policy has either a switching or an exhaustive structure. Ahn et al. (2004), Bell and Williams (2001, 2005), Harrison and López

(1999), Mandelbaum and Stolyar (2004), and Williams (2000) are among the papers that study flexible servers in parallel queues (they either consider a clearing system Ahn et al. (2004), or assume infinite buffer capacity and analyze the system in heavy traffic (Bell and Williams, 2001, 2005; Harrison and López, 1999; Mandelbaum and Stolyar, 2004; Williams, 2000).

Some papers study throughput maximization in tandem lines with finite buffers and flexible servers. Andradóttir et al. (2001) characterize the optimal server assignment policy for Markovian systems with $M = N = 2$ and provide near-optimal heuristic server assignment policies for larger systems. In a more recent work, Arumugan et al. (2008) study the optimal dynamic assignment of servers in a Markovian tandem system with $M = N = 2$ and a finite supply of jobs. Andradóttir and Ayhan (2005) characterize the optimal server assignment policy for Markovian systems with $M > N = 2$, and Andradóttir et al. (2008) consider the effects of server failures in the same settings. Kırkızlar et al. (2008) show that policies found to be optimal or near-optimal for Markovian systems in Andradóttir et al. (2001) and Andradóttir and Ayhan (2005) are also effective in non-Markovian systems. Andradóttir et al. (2003, 2007a) study a general queueing network with infinite buffers, without or with server and station failures, respectively. Other work on flexible servers in more general queueing systems includes Hajek (1984) and Tassiulas and Bhattacharya (2000).

Some papers focus on line balancing via server flexibility. Bartholdi and Eisenstein (1996a) show that in a line with servers ordered from slowest to fastest, infinitely divisible jobs, and deterministic service times, the "bucket brigade" policy results in the maximum achievable throughput, as well as a stable partition of work. Bartholdi et al. (1999) study the long-run behaviour of bucket brigades, concentrating on two- and three-worker lines. Bartholdi et al. (2001) study the performance of the bucket brigade policy when work consists of discrete tasks with exponentially distributed service requirements. Lim and Yang (2009) consider bucket brigade policies in understaffed lines with discrete work stations under the assumptions that the intermediate buffers have zero capacity, service times are deterministic, and servers cannot collaborate on the same task. Ahn and Righter (2006), Gel et al. (2002), McClain et al. (1992), Ostolaza et al. (1990), and Zavadlav et al. (1996) also consider the line-balancing problem in various tandem systems.

Finally, we review papers that compare the benefits of partial flexibility with full flexibility. Most of the work on partial flexibility consider parallel systems. Jordan and Graves (1995) study a setting with multiple products and plants, and show that most of the demand can be satisfied even with partially flexible plants (as opposed to plants that can produce all the products), as long as the assignment of products to plants is done well. Graves and Tomlin (2003) study a similar problem in multi-stage supply chains, and show that certain partial flexibility structures (chains, to be more specific) in each stage are sufficient, and that there is no need to coordinate the flexibility structures in different stages. Sheikhzadeh et al. (1998) study the assignment of products to machines in a plant and consider operational issues such as finite storage spaces, setup times, work-in-process (WIP), inventory levels, and manufacturing lead-times. Their work also supports the conclusion that most of the benefits of full flexibility can be obtained by partial flexibility (using chaining structures). Gurumurthi and Benjaafar (2004) study a parallel service system with flexible servers. They show that asymmetric server allocations are generally better than chaining structures if the servers are heterogeneous and different customer types have different demand rates. Wallace and Whitt (2005) study routing and server assignment in a call center, and show that most of the benefits of full flexibility can be reached even with one additional skill per agent. For a queueing network with outside arrivals and infinite buffers, Andradóttir et al. (2003) show that partial flexibility is sufficient for achieving the maximal capacity of the system.

There are also some papers that study the benefits of partial flexibility in tandem systems. Andradóttir et al. (2007b) provide numerical results showing that for systems with two stations, generalist servers, and exponentially distributed service times, most of the benefits of full flexibility can be attained with only one flexible server when the buffer size is sufficiently large. Hopp et al. (2004) study the capacity balancing problem for a line with equal number of workers and stations under a CONWIP (constant work-in-process) policy, and show that a skill-chaining strategy with two skills per worker outperforms a "cherry picking" strategy in which some workers are cross-trained at bottleneck stations, especially in systems with high variability and low WIP. However, Andradóttir et al. (2007b) and Hopp et al. (2004) only consider lines with equal number of servers and stations, or more servers than stations. By contrast, in this work we consider understaffed tandem lines and study all possible flexibility structures. We also consider a longer line and more general service rates compared to Andradóttir et al. (2007b). Moreover, we have a different objective, release policy, and collaboration structure than Hopp et al. (2004).

The remainder of the paper is organized as follows. In Section 2, we formulate the problem and provide preliminary results. In Section 3, we consider systems with deterministic service requirements, three stations, two flexible servers, and different flexibility structures, and identify both the optimal assignment policy and also the critical skills sufficient to attain the benefits of full flexibility. In Section 4, we analyze the corresponding Markovian system with small buffer sizes and identify the optimal server assignment policy for different flexibility structures. In Section 5, we propose heuristic server assignment policies based on the insights obtained from the optimal policies found in Sections 3 and 4, show these simple server assignment rules can achieve near-optimal throughput in Markovian systems with larger buffer sizes, and indicate how the optimal (partial) flexibility structure for deterministic, infinite-buffered systems can be used to select an effective (partial) flexibility structure for stochastic systems with finite buffers. In Section 6, we make some concluding remarks.

2 Problem Formulation

We consider tandem lines with N stations, M flexible servers, and finite intermediate buffers. We let $0 \leq \mu_{ij} < \infty$ denote the deterministic rate with which server $i \in \{1, \ldots, M\}$ works at station $j \in \{1, \ldots, N\}$. Without loss of generality, we assume that $\sum_{j=1}^{N} \mu_{ij} > 0$ for $i \in \{1, \ldots, M\}$, because otherwise some servers are not trained at any task, and this is equivalent to the case with fewer servers. Furthermore, we assume that $\sum_{i=1}^{M} \mu_{ij} > 0$ for $j \in \{1, \ldots, N\}$, because otherwise there is a station where nobody is trained at, and hence any policy will result in zero production rate. Several servers can collaborate on the same job, and their service rates are additive in this case. Service times at each station $j \in \{1, \ldots, N\}$ are independent and identically distributed with mean $m(j)$, and service times at different stations are independent. Without loss of generality, we assume that $m(j) = 1$ for all $j \in \{1, \ldots, N\}$. Travel times of the servers and setup times at the stations are negligible.

Let the state space S of the system be chosen to keep track of the number of jobs at each station and the status (operating, starved, or blocked) of each station. We say that a station is "operating" if that station is neither starved nor blocked. Let Π be the set of Markovian stationary deterministic policies corresponding to S. For all $\pi \in \Pi$ and $t \geq 0$, let $D^{\pi}(t)$ be the number of departures under policy π by time t, and let

$$T^{\pi} = \limsup_{t \to \infty} \frac{\mathbb{E}[D^{\pi}(t)]}{t}$$

be the long-run average throughput corresponding to the server assignment policy π. We are interested in solving the optimization problem

$$\max_{\pi \in \Pi} T^{\pi}. \tag{1}$$

For any $s \in S$, let A_s denote the set of allowable actions in state s. Possible actions are idling a server or assigning the server to station 1, 2, or 3. The term "idling" refers to voluntary idling of a server (as opposed to assigning a server to a station where (s)he is unable to work).

Specifically, for systems with three stations, for all $\pi \in \Pi$, we use the stochastic process $\{X^\pi(t) = (X_1^\pi(t), X_2^\pi(t)) : t \geq 0\}$, where $X_1^\pi(t)$ ($X_2^\pi(t)$) is the number of jobs that have already been processed at station 1 (2) and are either waiting for service or being processed at station 2 (3) at time $t \geq 0$. More specifically, for all $t \geq 0$, $X_1^\pi(t) = 0$ ($X_2^\pi(t) = 0$) if station 2 (3) is starved; $X^\pi(t) = (B_2 + 1, B_3 + 2)$ or $X_1^\pi(t) = B_2 + 2$ if station 1 is blocked; $X_2^\pi(t) = B_3 + 2$ if station 2 is blocked; $X^\pi(t) = (s_1, s_2)$ for $s_1 \in \{1, \ldots, B_2 + 1\}$, $s_2 \in \{1, \ldots, B_3 + 1\}$ if there are jobs that are being processed at both stations, $s_1 - 1$ jobs waiting in the buffer between stations 1 and 2, and $s_2 - 1$ jobs waiting in the buffer between stations 2 and 3. Consequently,

$$S = \{(s_1, s_2) : \ s_1 \in \{0, 1, \ldots, B_2 + 2\}, \ s_2 \in \{0, 1, \ldots, B_3 + 2\}, \ s_1 + s_2 \leq B_2 + B_3 + 3\}$$

is the state space.

The following lemma is a generalization of Corollaries 2.1 and 2.2 of Kırkızlar et al. (2008) to the system with two servers and more than two stations, and its proof follows directly from the proof of Lemma 2.1 of Kırkızlar et al. (2008).

Lemma 2.1 *For a tandem line with $N > 2$ and $M = 2$, there exists an optimal policy that is non-idling.*

3 Deterministic Systems

In this section we determine optimal server assignment policies for systems with deterministic service times. We also show that partial server flexibility can attain the throughput of full flexibility, and *provide the critical skills needed to achieve this goal.* Consider the following "allocation" linear program (LP) with decision variables λ and $\{\delta_{ij}\}$:

$$\left.\begin{array}{ll} \max & \lambda \\ \text{s.t.} & \delta_{1j}\mu_{1j} + \delta_{2j}\mu_{2j} \geq \lambda, \ \text{for } j \in \{1,2,3\}; \\ & \delta_{i1} + \delta_{i2} + \delta_{i3} \leq 1, \ \text{for } i \in \{1,2\}; \\ & \delta_{ij} \geq 0, \ \text{for all } i \in \{1,2\} \text{ and } j \in \{1,2,3\}. \end{array}\right\} \quad (2)$$

Let λ^* denote the optimal value of λ for this LP. Andradóttir et al. (2003) show that λ^* is the maximal capacity of an infinite-buffered tandem line with three stations, two flexible servers, and outside arrivals. Lemma 2.3 of Kırkızlar et al. (2008) shows that λ^* is an upper bound on the throughput of our finite-buffered tandem line as well. Moreover, if $\{\delta_{ij}^*\}$ are optimal values of $\{\delta_{ij}\}$, then δ_{ij}^*, where $i \in \{1,2\}$ and $j \in \{1,2,3\}$, can be interpreted as the long-run proportion of time server i should be assigned to station j in order to achieve the maximal capacity λ^*.

The LP described in the previous paragraph has five constraints (in addition to the nonnegativity constraints) and seven variables. Since λ is always positive, we can conclude that at least two elements of the set $\{\delta_{ij}\}$ are zero (this also follows directly from Proposition 2 of Andradóttir et al., 2003). This proves that four skills are sufficient to achieve the maximal capacity in systems with infinite buffers, and hence it is of interest to determine to what extent this is also true for systems with finite buffers. Consequently, we first analyze the system under the assumption that the two servers have a total of four skills, and then show the implications of this result for systems with fully cross-trained servers.

Without loss of generality, we assume that the system initially starts in a state $s^0 = (s_1^0, s_2^0)$ where all the stations are operating. We let $S^0 \subset S$ be the set of such states. If the system does not start in a state in S^0, initially any policy that takes the process to such a state may be employed. For example, we can successively assign all servers to the station $j \in \{2, \ldots, N\}$ that is closest to the end of the line among the stations that are operating. When there are no such stations left, we can assign all servers to the station $j \in \{1, \ldots, N-1\}$ that is closest to the beginning of the line among the stations that are preceding a station that is starved. When this is no longer possible (because there are no such stations left), the system is in a state $s^0 \in S^0$ with remaining service time equal to one at all stations. This is achievable in finite time and will not affect the long-run average throughput.

Let $s^0 \in S^0$ and let $\mathcal{L}_1(\mathcal{L}_2)$ be an ordered list of preferred stations for server 1 (2). We define the server assignment policy $\pi_{s^0,\mathcal{L}_1,\mathcal{L}_2}$ as follows:

> Every time the system reaches state s^0, assign server 1 (2) to the stations on the list $\mathcal{L}_1$ ($\mathcal{L}_2$) in order until one job is completed at each station in the system.

In other words, each server is originally assigned to the first station on its list. When a server completes work at a station on its list, the server checks if the system has returned to state s^0. If the answer is yes, the server starts work at the first station on its list; if the answer is no, the server will either idle or move to the next station on its list depending on whether the server has or has not reached the end of its list.

We consider systems with one dedicated and one fully flexible server in Section 3.1. Then, we study systems with two partially flexible servers in Section 3.2. Finally, in Section 3.3, we determine the critical skills needed to achieve the optimal performance of systems with fully flexible servers.

3.1 Systems with One Dedicated and One Fully Flexible Server

In this section, we provide a theorem identifying the optimal server assignment policies for systems with one dedicated and one fully flexible server. Without loss of generality, we assume that the first server is the dedicated server, since the other case is equivalent to this one by relabeling the servers. The proof of Theorem 3.1 is provided in Appendix A.

Theorem 3.1 *Consider a deterministic system with three stations and two servers. Assume that server 1 is dedicated to station* $D \in \{1, 2, 3\}$*, server 2 is fully flexible, and the system is initially in a state* $s^0 \in S^0$*. Let* k_1 *(*k_2*) denote the station that is the closest (second closest) station to station* D *(with ties broken arbitrarily),* $\mathcal{L}_1 = \{D\}$*, and* $\mathcal{L}_2 = \{k_1, k_2, D\}$*. Then* $\pi_{s^0,\mathcal{L}_1,\mathcal{L}_2}$ *is optimal and attains the maximal system capacity* λ^**, regardless of the intermediate buffer sizes.*

Note that the policy of Theorem 3.1 is optimal among all possible policies (not only over Π) because it attains the maximal capacity of the system. Furthermore, it is possible to attain the maximal capacity of the four-skilled system above even with three skills (i.e., with $\mu_{2D} = 0$) when $\mu_{1D} \geq \mu_{2k_1}\mu_{2k_2}/(\mu_{2k_1} + \mu_{2k_2})$.

We now discuss the three special cases covered by Theorem 3.1 in more detail, namely when server 1 is dedicated to station 1, 2, or 3, respectively. When $D = 1$ (i.e., $\mu_{12} = \mu_{13} = 0$), the flexible server starts working at station 2, moves to station 3 upon completion of the task at station 2, and finally moves to station 1. Moreover, the dedicated server idles only if his/her service rate is high enough that utilizing him/her more would cause blocking at the first station, rather than increasing the throughput. The flexible server helps the dedicated server at station 1 if (s)he is fast enough to complete jobs at stations 2 and 3 before the first server completes one job at station 1. When $D = 3$, the optimal policy is "symmetrical" with respect to the case where $D = 1$. Now, the flexible server starts working at station 2, moves to station 1 upon completion of the task at station 2, and finally moves to station 3. Idling of the dedicated server also occurs only if his/her service rate is high enough that utilizing him/her more would cause starvation at the third station, and not increase the throughput. Finally, when $D = 2$, similar to the previous cases, we see that the dedicated server is idle only if his/her service rate is high enough that (s)he would not increase the throughput by being utilized more. Moreover, by idling this server at certain times, we are able to keep both stations operating rather than causing starvation or blocking at the second station.

We observe that when there is a dedicated and a fully flexible server, the optimal assignment for the flexible server focuses on keeping the dedicated server's station operating at all times.

By giving the priority to the stations that provide jobs for or process jobs from the dedicated server's station, we ensure that the dedicated server is not idling. When the dedicated server is at station 2, the policies that have the flexible server giving preference to prevent either blocking or starvation of station 2 are both optimal. Moreover, when the dedicated server is at station 1 (3), one can use similar arguments as in the proof of Theorem 3.1 to show that the policy that assigns the flexible server to station 3 (1) before station 2 in every regenerative cycle is also optimal. Hence, without any efficiency loss, the flexible server can process the jobs in arbitrary order at the stations where there is no dedicated server when the service times are deterministic.

3.2 Systems with Two Partially Flexible Servers

In this section, we consider systems where each server is partially cross-trained; i.e., each server is capable of processing jobs at two stations. The proof of the following theorem is similar to that of Theorem 3.1, and hence is omitted here (however, it is provided in Kırkızlar, 2008).

Theorem 3.2 *Consider a deterministic system with three stations and two servers. Assume that both servers are partially flexible and that the system is initially in a state $s^0 \in S^0$. Let F denote the station where both servers are cross-trained, F_1 (F_2) be the station where only server 1 (2) is cross-trained, $\mathcal{L}_1 = \{F_1, F\}$, and $\mathcal{L}_2 = \{F_2, F\}$. Then $\pi_{s^0,\mathcal{L}_1,\mathcal{L}_2}$ is optimal and attains the maximal system capacity λ^*, regardless of the intermediate buffer sizes.*

Note that the policy of Theorem 3.2 is optimal among all possible policies (not only over Π) because it attains the maximal capacity of the system. Further, the maximal capacity of the four-skilled system can be reached even with three skills when either $\mu_{1F_1} \leq \mu_{2F}\mu_{2F_2}/(\mu_{2F}+\mu_{2F_2})$ or $\mu_{2F_2} \leq \mu_{1F}\mu_{1F_1}/(\mu_{1F} + \mu_{1F_1})$. In the former case, the optimal throughput can be achieved with $\mu_{1F} = 0$; in the latter case, the optimal throughput can be achieved with $\mu_{2F} = 0$.

The description of the optimal policy in Theorem 3.2 shows that each server starts working at the station where (s)he is the only server trained to work. Then, after completing the job at the station they are primarily assigned to, the servers move to the other station where they are both trained to work. Thus, this assignment policy gives priority to the tasks that can be done by only one server and ensures that servers are neither starved nor blocked. Furthermore, idling occurs only when one server is so fast that (s)he can complete one job at two stations before the other server completes a job at one station. In this case, we idle the fast server in order to balance the line and keep all the stations operating, because utilizing the fast server more causes starvation or blocking in the system but does not increase the throughput. Perfect coordination of the servers in order to prevent any productivity loss is achievable because the service times are deterministic.

3.3 Identifying the Best Flexibility Structure

Theorems 3.1 and 3.2 show that when the servers have four skills and the service times are deterministic, it is possible to reach the maximal capacity of the corresponding four-skilled infinite-buffered systems in the finite-buffered settings. In this section, given the potential skill of each server at each task (if the server were trained to perform the task), we determine the four critical skills that are sufficient to attain the maximal capacity of the fully flexible system. We need the following conditions:

$$\begin{array}{llll}
\{1\}\ \mu_{11}\mu_{22} \geq \mu_{12}\mu_{21}; & \{2\}\ \mu_{11}\mu_{22} < \mu_{12}\mu_{21}; & \{3\}\ \mu_{11}\mu_{23} \geq \mu_{13}\mu_{21}; & \{4\}\ \mu_{11}\mu_{23} < \mu_{13}\mu_{21}; \\
\{5\}\ \mu_{12}\mu_{23} \geq \mu_{13}\mu_{22}; & \{6\}\ \mu_{12}\mu_{23} < \mu_{13}\mu_{22}; & \{7\}\ \mu_{11} \leq \dfrac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}; & \{8\}\ \mu_{11} > \dfrac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}; \\
\{9\}\ \mu_{12} \leq \dfrac{\mu_{21}\mu_{23}}{\mu_{21}+\mu_{23}}; & \{10\}\ \mu_{12} > \dfrac{\mu_{21}\mu_{23}}{\mu_{21}+\mu_{23}}; & \{11\}\ \mu_{13} \leq \dfrac{\mu_{21}\mu_{22}}{\mu_{21}+\mu_{22}}; & \{12\}\ \mu_{13} > \dfrac{\mu_{21}\mu_{22}}{\mu_{21}+\mu_{22}}; \\
\{13\}\ \mu_{21} \leq \dfrac{\mu_{12}\mu_{13}}{\mu_{12}+\mu_{13}}; & \{14\}\ \mu_{21} > \dfrac{\mu_{12}\mu_{13}}{\mu_{12}+\mu_{13}}; & \{15\}\ \mu_{22} \leq \dfrac{\mu_{11}\mu_{13}}{\mu_{11}+\mu_{13}}; & \{16\}\ \mu_{22} > \dfrac{\mu_{11}\mu_{13}}{\mu_{11}+\mu_{13}}; \\
\{17\}\ \mu_{23} \leq \dfrac{\mu_{11}\mu_{12}}{\mu_{11}+\mu_{12}}; & \{18\}\ \mu_{23} > \dfrac{\mu_{11}\mu_{12}}{\mu_{11}+\mu_{12}}. & &
\end{array}$$

Conditions $\{1\}$ through $\{6\}$ compare the relative speeds of servers at different stations. For example, condition $\{1\}$ implies that server 1 is relatively faster at station 1 than server 2 (at the same time server 2 is relatively faster at station 2 than server 1). Conditions $\{7\}$ through $\{18\}$ compare the service completion rate of servers in different zones. For example, condition $\{7\}$ implies that the service completion rate of server 2 in the zone consisting of stations 2 and 3 is higher than the service completion rate of server 1 at station 1.

The following theorem, whose proof is provided in Appendix B, specifies the best flexibility structure for a system with three stations and two servers.

Theorem 3.3 *For a tandem line with three stations, two flexible servers, arbitrary buffer sizes between the stations, and deterministic service times, the assignment (and hence cross-training) policy specified in Table 1 is optimal.*

Note that Theorem 3.3 employs the optimal solution of the allocation LP, and hence it also provides the optimal assignment policy for the corresponding infinite-buffered system. Next, we show that any set of service rates μ_{ij}, where $i \in \{1, 2\}$ and $j \in \{1, 2, 3\}$, has to satisfy exactly one of the cases mentioned in Theorem 3.3. The proof of this result is provided in Appendix C.

Table 1: Critical Skills and Optimal Server Assignment Policy for Small Deterministic Systems

Case	*Conditions Satisfied*	*Optimal Server Assignment Policy*
a	{*1*}, {*3*}, {*7*}	*Let* $D = 1$ *and use Theorem 3.1*
b	{*2*}, {*4*}, {*13*}	*Relabel servers, let* $D = 1$, *and use Theorem 3.1*
c	{*2*}, {*5*}, {*9*}	*Let* $D = 2$ *and use Theorem 3.1*
d	{*1*}, {*6*}, {*15*}	*Relabel servers, let* $D = 2$, *and use Theorem 3.1*
e	{*4*}, {*6*}, {*11*}	*Let* $D = 3$ *and use Theorem 3.1*
f	{*3*}, {*5*}, {*17*}	*Relabel servers, let* $D = 3$, *and use Theorem 3.1*
g	{*2*}, {*3*}, {*10*}, {*18*}	*Let* $F = 1$ *and use Theorem 3.2*
h	{*1*}, {*4*}, {*12*}, {*16*}	*Relabel servers, let* $F = 1$, *and use Theorem 3.2*
i	{*1*}, {*5*}, {*8*}, {*18*}	*Let* $F = 2$ *and use Theorem 3.2*
j	{*2*}, {*6*}, {*12*}, {*14*}	*Relabel servers, let* $F = 2$, *and use Theorem 3.2*
k	{*3*}, {*6*}, {*8*}, {*16*}	*Let* $F = 3$ *and use Theorem 3.2*
l	{*4*}, {*5*}, {*10*}, {*14*}	*Relabel servers, let* $F = 3$, *and use Theorem 3.2*

Proposition 3.1 *The twelve cases* $\{a, \ldots, l\}$ *in Theorem 3.3 are mutually exclusive and collectively exhaustive.*

The criteria provided in Table 1 for deciding the best flexibility structure can be summarized as follows. If one server is relatively fast at one station with respect to the other stations (for example, conditions $\{1\}$ and $\{3\}$ imply that server 1 is relatively fast at station 1 in case a) and (s)he cannot finish one job at that station before the other server finishes service at both of the other stations (condition $\{7\}$ in case a), then this server should be dedicated to the station where (s)he is relatively fast. On the other hand, if the two servers are relatively fast at different stations with respect to the same station (for example, in case g, conditions $\{2\}$ and $\{3\}$ imply that server 1 is relatively better at station 2 compared to station 1 and server 2 is relatively better at station 3 compared to station 1) and they can finish a job at the station they are relatively fast at before the other server can finish service at both of the other stations (conditions $\{10\}$ and $\{18\}$ in case g), then they should work at the station where they are relatively fast at, and also at the common station where they are both relatively slow. The other cases (b through f and h through l) can be described similarly by simply changing the labeling of the servers and the stations they are relatively faster or slower at.

In summary, we have proved that the optimal cross-training strategy in a finite-buffered system with deterministic service times is the same as the one of the corresponding infinite-buffered system. Moreover, the maximal possible throughput (corresponding to full cross-training and infinite buffers) can be obtained with partial cross-training and finite buffers for deterministic systems, regardless of the size of the buffers.

4 Markovian Systems

In this section, we consider systems with three stations, two servers, and exponentially distributed service requirements at each station. Theorems 8.1.2 and 9.1.8 of Puterman (1994) show the existence of an optimal Markovian stationary deterministic policy because the state space S and action space A_s, where $s \in S$, are finite. Hence, our assumption that Π consists of Markovian stationary deterministic policies is not restrictive.

For $i = \{1, 2\}$, let $\mathcal{L}_i = \{\mathcal{L}_i(s) : s \in S\}$ be a set of ordered lists of preferred stations for server i. Let $\pi^e = (d^e_{\mathcal{L}_1,\mathcal{L}_2})^\infty$, where for all $s \in S$, the action $d^e_{\mathcal{L}_1,\mathcal{L}_2}(s)$ is defined as follows:

> When the system is in the state s, assign server 1 (2) to the first station that is operating (neither starved nor blocked) in $\mathcal{L}_1(s)$ ($\mathcal{L}_2(s)$).

In Section 4.1, we present our observations about the form of the optimal server assignment policy based on numerical experiments. In Section 4.2, we show that four-skilled systems attain near-optimal throughput as compared to fully cross-trained systems. Finally, we identify optimal server assignment policies for systems with one dedicated and one fully flexible server in Section 4.3, and for systems with two partially flexible servers in Section 4.4.

4.1 Fully Cross-Trained Servers

When both servers are cross-trained at all the stations (i.e., $\mu_{ij} > 0$ for all $i \in \{1, 2\}$ and $j \in \{1, 2, 3\}$), the optimal server assignment policy is difficult to characterize. Even for systems with fewer skills and small buffer sizes, we observe in Theorems 4.1 and 4.2 below that the optimal policy can be somewhat complex. The optimal assignment policy for fully flexible systems appears to be more complicated than these, and is also difficult to implement in real life. Hence, we performed simulation experiments to determine the optimal assignment of fully cross-trained servers. We randomly generated 10,000 systems where the service rates were drawn independently from a uniform distribution with range [0.5,2.5]. Then, assuming that $B_2 = B_3 = B$, we used the policy iteration algorithm for communicating Markov chains to identify the optimal server assignment policy for each system and each $B \in \{0, 1, 2, 3, 4\}$. Here are our observations:

- At least one of the servers (sometimes both of them) appears to have a primary assignment. In other words, at least one server is assigned to a specific station as long as that station is neither starved nor blocked.
- Primary assignment can change with the buffer size. In other words, sometimes a server that has a primary assignment for one buffer size either does not have a primary assignment or has a different primary assignment for another buffer size.
- Primary assignment is not always where the server is fastest or according to a simple multiplicative rule (which is the case when there are two stations in tandem and two flexible servers, as shown in Andradóttir et al., 2001).
- If both servers have primary assignments, their primary assignment is not at the same station.

We observe in Sections 4.3 and 4.4 that some of the conclusions above also hold for the optimal assignment policy for partially flexible servers. More specifically, we will see that at least one server in lines with limited flexibility and small buffers has a primary assignment, and we will identify the primary assignment of such servers.

4.2 Partially Cross-Trained Servers

When the service requirements at each station are deterministic, it is shown in Section 3.3 that the maximum throughput (corresponding to fully trained servers) can be achieved when the servers are cross-trained to have four skills in total. In Markovian systems, it is not possible to reach such a conclusion because of the stochastic nature of the problem. Nevertheless, there is strong evidence that near-optimal throughput can be obtained with four skills only. We performed 50,000 experiments for the systems with the same parameters as in Section 4.1. We found the maximum throughput of all three-skilled (i.e., systems that have only one server trained at each station) and four-skilled systems and compared it to the maximum throughput of the system with six skills (i.e., both servers are fully cross-trained). The average performance of the best three-skilled and four-skilled systems, given as a percentage of the optimal performance of the six-skilled system, is listed in Table 2 for $B_2 = B_3 \in \{0, 1, \ldots, 4\}$.

We conclude that even for small buffer sizes, it is possible to achieve near-optimal throughput (compared to that of the fully flexible system) with only four skills. Moreover, the benefit of adding one more skill to the system is obvious when we compare the three-skilled and four-skilled systems

Table 2: Comparison of Best Three-Skilled and Four-Skilled Systems with Six-Skilled Systems

Buffer Sizes	Best 3-skilled	Best 4-skilled
$B_2 = B_3 = 0$	74.65%	91.94%
$B_2 = B_3 = 1$	79.08%	96.67%
$B_2 = B_3 = 2$	81.00%	98.27%
$B_2 = B_3 = 3$	81.83%	99.01%
$B_2 = B_3 = 4$	82.59%	99.37%

(more specifically, by adding one more skill to the system, the optimality gap closes by at least 68% and sometimes by more than 96%). Hence, it is important to identify the optimal assignment policy for systems with four skills. We start by determining the optimal server assignment policy for systems with one dedicated and one fully flexible server in Section 4.3. Then, we consider systems with two partially flexible servers in Section 4.4. We limit ourselves to small buffer sizes because the expressions become highly complex for larger buffer sizes.

4.3 Systems with One Dedicated and One Fully Flexible Server

In this section, we identify the optimal server assignment policy when one server is dedicated at stations 1, 2, or 3, respectively, and the other server is cross-trained at all stations. Without loss of generality, we assume that the first server is the dedicated server because otherwise we can relabel the servers. The proof of Theorem 4.1 is provided in Appendix D.

Theorem 4.1 *Consider a Markovian system with three stations and two servers. Assume that server 1 is dedicated to station D, server 2 is fully flexible, and $B_2, B_3 \leq 1$. Let k_1 (k_2) denote the station that is closest (second closest) to station D (with ties broken arbitrarily), and define*

$$S' = \{s \in S : s = (s_1, B_3 + 1) \text{ for } s_1 > 1, \text{ or } s = (s_1, B_3 + 2) \text{ for } s_1 > 0\}.$$

Let $\mathcal{L}_1(s) = \{D\}$ for all $s \in S$, and let $\mathcal{L}_2(s)$ be defined as follows:

- *If $D \in \{1, 3\}$, then $\mathcal{L}_2(s) = \{k_1, k_2, D\}$ for all $s \in S$.*
- *If $D = 2$, then $\mathcal{L}_2(s) = \{3\}$ for all $s \in S'$ and $\mathcal{L}_2(s) = \{1, 3, 2\}$ for all $s \in S \backslash S'$.*

Then $\pi^e = (d^e_{\mathcal{L}_1,\mathcal{L}_2})^\infty$ is optimal in Π.

We observe that if there is one dedicated and one flexible server, the flexible server does not work at the station where the dedicated server is working, as long as there is another operating station, to avoid idling the dedicated server. The assignment of the flexible server when both of the other stations are operating has the goal of keeping the dedicated server's station operating.

When $D \in \{1, 3\}$, the optimal server assignment policy is similar to that of the system with deterministic service times. When the dedicated server is at station 1 (3), the main goal is to prevent blocking (starvation) at station 1 (3), and hence the flexible server gives priority to station 2, then to station 3 (1), and finally moves to the station where the dedicated server is working. This prioritization is the same as in the corresponding deterministic systems.

When the dedicated server is at station 2, two goals (to prevent starvation and blocking) conflict with each other. This explains the more complex structure of the optimal assignment of the flexible server when $D = 2$ in Theorem 4.1. In order to keep station 2 operating, the optimal policy gives priority to station 1 unless $s \in S'$, where station 2 is either blocked (but not starved) or about to be blocked (but not starved); in such states the optimal policy gives priority to station 3. After station 1, the flexible server gives the second highest priority to station 3, and station 2 is the least preferred station. The optimal policy for the corresponding deterministic system (given in Theorem 3.1) also has a similar structure in that either of stations 1 or 3 can be given priority, but station 2 is the least preferred station. Note that the local heuristic in Andradóttir et al. (2001) gives preference to removing blocking rather than starving in longer lines. We see that the optimal policy in our system puts higher priority on removing starving than blocking, but it also considers the immediate blocking possibility in station 2 and tries to prevent blocking before it even happens.

We conclude this section by pointing out that the optimal policy provided in Theorem 4.1 is not necessarily unique. For example, the proof of Theorem 4.1 in Appendix D for systems with $D = 1$ and $B_2 = B_3 = 0$ suggests that the actions a_{12} and a_{13} are both optimal whenever these actions are both in A_s. However, when $B_2 = B_3 = 1$, there are some states s with $a_{12}, a_{13} \in A_s$ where a_{12} is strictly better than a_{13}. Hence, the policy descriptions in the theorem were chosen so that the same policy would be optimal for systems with different buffer sizes.

4.4 Systems with Two Partially Flexible Servers

In this section, we consider four-skilled systems where each server is cross-trained to work at two stations. The following theorem provides the optimal server assignment policy under different cross-training strategies. We limit ourselves to the systems with zero buffer sizes, because for larger buffer sizes the expressions become highly complex and the optimal policy is difficult to characterize. The proof of the following theorem is similar to that of Theorem 4.1, and hence is omitted here (however, it is provided in Kırkızlar, 2008).

Theorem 4.2 *Consider a Markovian system with three stations and two servers. Assume that both servers are partially flexible and $B_2 = B_3 = 0$. Let F denote the station where both servers are cross-trained, F_1 (F_2) be the station where only server 1 (2) is cross-trained, and assume that the servers are labeled so that $F_1 < F_2$. Define*

$$S'' = \{s \in S : s = (0, s_2) \text{ for } s_2 < B_3 + 2\}.$$

For all $s \in S$, let $\mathcal{L}_1(s)$ and $\mathcal{L}_2(s)$ be defined as follows:

- *If $F = 2$ and $\mu_{11}^2\mu_{12}^2 > \mu_{22}\mu_{23}(\mu_{11}\mu_{12} + \mu_{11}\mu_{23} + \mu_{12}\mu_{23} + \mu_{23}^2)$, then $\mathcal{L}_1(s) = \{F, F_1\}$; otherwise $\mathcal{L}_1(s) = \{F_1, F\}$.*
- *If $F = 1$, $\mu_{11}\mu_{12} < \mu_{21}\mu_{23}$, and $s \in S''$, then $\mathcal{L}_2(s) = \{F\}$; otherwise $\mathcal{L}_2(s) = \{F_2, F\}$.*

Then $\pi^e = (d^e_{\mathcal{L}_1,\mathcal{L}_2})^\infty$ is optimal in Π.

When both servers are cross-trained at station 1, server 1 (who is cross-trained at stations 1 and 2) has a primary assignment at station 2. When $\mu_{11}\mu_{12} \geq \mu_{21}\mu_{23}$ (which can be interpreted as server 1 having better overall performance), server 2 has a primary assignment at station 3. This is reasonable because server 1 is already performing well at stations 1 and 2, and the capacity of server 2 can be primarily given to station 3. When $\mu_{11}\mu_{12} < \mu_{21}\mu_{23}$, server 2 does not have a primary assignment at any station, but gives priority to station 3 except when station 2 is starved but not blocked (so that $s \in S''$). In this case, since server 2 performs well at station 3, (s)he can shift some capacity to station 1 without causing poor performance at station 3. This also allows the slower server (server 1 in this case) to spend more time on the task where the faster server cannot work. Hence, we conclude that the focus of server 2 depends on how the performance of server 1 compares to his/her own, with the measure of performance of each server being the product of the servers' rates at the tasks they are trained for ($\mu_{11}\mu_{12}$ and $\mu_{21}\mu_{23}$).

By symmetry, when both servers are cross-trained at station 3, we would expect server 1 to give priority to station 1 and move to station 3 when station 2 is blocked but not starved, for some service rates. However, Theorem 4.2 shows that when $F = 3$, both servers have primary assignments at the stations where only one server is cross-trained to work. Nevertheless, closer examination suggests that the policies are more symmetrical than it first appears because when

$B_2 = B_3 = 0$, $(1,2)$ is the only state where station 2 is blocked but not starved. In fact, station 1 is also blocked in this state, and server 1 moves to station 3 under the policy of Theorem 4.2.

The symmetry between the cases where both servers are trained at station 1 or 3, respectively, can be observed in systems with $(B_2, B_3) \neq (0,0)$. For example consider the case where $\mu_{11} = 2$, $\mu_{12} = 0$, $\mu_{13} = 3$, $\mu_{21} = 0$, $\mu_{22} = 1$, and $\mu_{23} = 1$ (so that $\mu_{11}\mu_{13} \geq \mu_{22}\mu_{23}$). When $B_2 = 1$ and $B_3 = 0$, the optimal policy assigns server 2 to station 2 if station 2 is operating, and to station 3 otherwise; and assigns server 1 to station 1 if station 1 is operating and station 2 is not blocked or both blocked and starved, and to station 3 otherwise. In other words, server 2 has a primary assignment at station 2, and the optimal assignment of server 1 is of threshold type. The optimal policy results in a throughput of 0.8472, but the policy of Theorem 4.2 for the case where $F = 3$ yields a throughput of 0.8100. Similarly, the optimal policy appears to be of threshold type when $\mu_{11}\mu_{12} \geq \mu_{21}\mu_{23}$ and $(B_2, B_3) \neq (0,0)$ in cases where both servers are trained at station 1.

When both servers are cross-trained at station 2, they both have primary assignments. Server 2 (who is cross-trained at stations 2 and 3) has a primary assignment at station 3 regardless of the service rates. However, server 1 can have a primary assignment at station 1 or 2. This shows a preference for clearing blocking in the system relative to starvation. If $\mu_{11}^2\mu_{12}^2 > \mu_{22}\mu_{23}(\mu_{11}\mu_{12} + \mu_{11}\mu_{23} + \mu_{12}\mu_{23} + \mu_{23}^2)$ holds, then server 1 has a primary assignment at station 2, and otherwise server 1 has a primary assignment at station 1. Unlike the corresponding condition in Theorem 4.2 for the case where $F = 1$, each side of this inequality does not consist of simple multiplication of the rates of each server at the different stations. The inequality seems to suggest which server has better overall performance, but in a way that skews the selection of the better overall server towards server 2 because the right-hand side is always bigger than $\mu_{22}\mu_{23}\mu_{11}\mu_{12}$. In other words, it is less likely that server 1 is primarily assigned to station 2 (rather than station 1) with this inequality than under the condition $\mu_{11}\mu_{12} > \mu_{22}\mu_{23}$ (which is the criterion in Theorem 4.2 for the case where $F = 1$ adapted to the current case).

Overall, we observe that in the four-skilled systems with two partially flexible servers, there is always one server with a primary assignment, and this result is consistent with what we observed for the fully-flexible system. For the small systems we considered, the primary assignment of one server does not depend on the service rates or buffer sizes. However, whether or not the other server has a primary assignment, and where (s)he is primarily assigned, may depend on the service rates and the buffer sizes. Moreover, the policies in Theorem 4.2 for the cases where $F = 1$ and $F = 3$ are symmetrical versions of each other (even though the special structure of the system with $B_2 = B_3 = 0$ makes them seem different), while the policy in Theorem 4.2 for the case where

$F = 2$ is different from the others (as expected). When both servers are cross-trained at station 2, the optimal policy is more complex than in the other cases, which may result from the fact that station 2 can be starved or blocked, and the assignment policy has to prevent both of these events to the extent possible.

Theorem 4.2 also shows that the optimal policy in the Markovian setting is slightly different from the optimal policy of the corresponding deterministic system. In the deterministic system, it is possible to coordinate the service completions so that no blocking or starvation occurs. Since this is not the case in the Markovian system, the optimal policy is of a "threshold" type that also aims to keep the stations operating. Furthermore, the form of the policy may be quite complicated, and the results for $B_2 = B_3 = 0$ do not generalize to systems with bigger buffer sizes (see also the numerical experiments in Section 5.1). Note also that the policy specified in Theorem 4.2 need not be unique (the proof of Theorem 4.2 in Kırkızlar, 2008, suggests that there may be multiple actions that are optimal in some states).

The optimal throughput can be calculated for the cross-training strategies presented in Theorems 4.1 and 4.2, but the task of finding the best partially flexible system for a given set of (potential) service rates (see Theorem 3.3) is not a simple task. When the expressions for the optimal throughputs are compared with each other, we obtain complex expressions that do not provide intuitive criteria to compare the flexibility structures. However, in the next section, we use the best flexibility structure for the corresponding deterministic system (see Theorem 3.3) and test the performance of the corresponding optimal policy (see Theorems 4.1 and 4.2) for small systems in larger Markovian systems. More specifically, we use the conditions in Table 1 to select a flexibility structure, and then use Theorem 4.k instead of Theorem 3.k, for $k \in \{1, 2\}$, to specify a server assignment policy.

5 Numerical Results

In this section, we provide both near-optimal heuristic server assignment policies and also guidelines for selecting good flexibility structures for understaffed Markovian lines. More specifically, in Section 5.1 we present and test server assignment heuristics for tandem lines with three stations, two servers, and four skills that were developed using the insights obtained from the optimal policies for deterministic and small Markovian systems provided in Sections 3 and 4, respectively. Then, in Section 5.2, we compare the performance of lines with limited and full flexibility, and show that the optimal flexibility structures for deterministic systems with three stations, two servers, and infinite buffers (obtained from the allocation LP, see Theorem 3.3) are also effective for the corresponding Markovian systems with finite buffers. Finally, in Section 5.3 we provide numerical results that suggest that the solution of the allocation LP also provides an effective flexibility structure for Markovian lines with more than three stations.

5.1 Heuristic Server Assignment Policies

In this section, we present and compare the following three heuristic server assignment policies for systems with three stations and two servers with a total of four skills between them.

Policy 1: Use policy $\pi^e = (d^e_{\mathcal{L}_1,\mathcal{L}_2})^\infty$, where $\mathcal{L}_1(s) = \mathcal{L}_1$ and $\mathcal{L}_2(s) = \mathcal{L}_2$ for all $s \in S$ and $\mathcal{L}_1$ ($\mathcal{L}_2$) is the best priority policy on average for server 1 (2).

Policy 2: The optimal assignment policy for Markovian systems with small buffer sizes (see Sections 4.3 and 4.4) is employed for systems with any buffer sizes.

Policy 3: A modification of Policy 2 that treats stations 1 and 3 symmetrically whenever this appears to yield improved performance.

Note that for some flexibility structures (i.e., when $D \in \{1, 3\}$ or $F = 3$), some of the three policies above are identical.

Policy 1 is motivated by the policies found to be optimal for deterministic systems with three stations, two servers, and four skills, see Theorems 3.1 and 3.2. In order to determine the priorities used in Policy 1, 50,000 systems were generated with service rates independently drawn from a uniform distribution with range [0.5,2.5] and the buffer sizes independently drawn from the discrete uniform distribution with range $\{0, 1, 2, 3, 4, 5\}$. Then, all possible assignments were compared and the one with the highest average throughput (computed using the stationary distribution of the corresponding Markov chain) in 50,000 experiments was selected. For all six flexibility structures under consideration, the best observed priority structure is optimal for the corresponding deterministic system (the only difference is that when $D = 2$, Policy 1 uses one of the two priority structures found to be optimal for deterministic systems).

We now compare Policy 1 with Policy 2. When $D \in \{1, 3\}$, the two policies coincide (see the corresponding cases in Theorem 4.1). When $D = 2$, we found that $\mathcal{L}_2 = \{1, 3, 2\}$ was the best priority policy (this agrees with Policy 2 except when $s \in S'$, see Theorem 4.1). When the servers have two skills, each server gives priority to the station where no other server is cross-trained, so that $\mathcal{L}_1 = \{F_1, F\}$ and $\mathcal{L}_2 = \{F_2, F\}$. Thus, Policy 1 is the same as Policy 2 when $F = 3$

(see Theorem 4.2), and is consistent with Policy 2 when $F \in \{1, 2\}$. We conclude that Policy 1 is either an equivalent or a simpler version of Policy 2, and that its simplicity makes Policy 1 appealing.

Next, we describe how we modified Policy 2 in order to treat stations 1 and 3 more symmetrically in Policy 3. For systems with one dedicated server and one fully flexible server, Policy 3 agrees with Policy 2 since no improvement over the optimal policy for small systems was found. Define

$$S''' = \{s \in S : s = (1, s_2) \text{ for } s_2 < B_3 + 1\},$$
$$S'''' = \{s \in S : s = (0, s_2) \text{ for } s_2 < B_3 + 2, \text{ or } s = (1, s_2) \text{ for } s_2 < B_3 + 1\}.$$

For systems with two partially flexible servers, Policy 3 is specified below:

- When $F = 1$, Policy 3 differs from Policy 2 only for $s \in S'''$ when $\mu_{11}\mu_{12} < \mu_{21}\mu_{23}$. In this case, Policy 3 assigns server 2 to station 1, but Policy 2 assigns server 2 to station 3.
- When $F = 3$, Policy 3 differs from Policy 2 only for $s \in S'$ when $\mu_{22}\mu_{23} < \mu_{11}\mu_{13}$. In this case, Policy 3 assigns server 1 to station 3, but Policy 2 assigns server 1 to station 1.
- When $F = 2$, Policy 3 differs from Policy 2 in the following cases. When $\mu_{11}^2\mu_{12}^2 > \mu_{22}\mu_{23}(\mu_{11}\mu_{12} + \mu_{11}\mu_{23} + \mu_{12}\mu_{23} + \mu_{23}^2)$ and for $s \in S'$, Policy 3 assigns server 1 to station 1, but Policy 2 assigns server 1 to station 2. When $\mu_{22}^2\mu_{23}^2 > \mu_{11}\mu_{12}(\mu_{22}\mu_{23} + \mu_{11}\mu_{23} + \mu_{11}\mu_{22} + \mu_{11}^2)$ and for $s \in S \backslash S''''$, Policy 3 assigns server 2 to station 2, but Policy 2 assigns server 2 to station 3.

We noted in Section 4.4 that the optimal policies for $F \in \{1, 3\}$ are in fact more symmetrical than it appears in Theorem 4.2 for the special case of $B_2 = B_3 = 0$. Moreover, the optimal policies for deterministic systems also treat stations 1 and 3 symmetrically (see Theorems 3.1 and 3.2). Hence, unlike Policy 2, Policy 3 for $F = 1$ and $F = 3$ are symmetric versions of each other, and Policy 3 for $F = 2$ treats stations 1 and 3 symmetrically.

To evaluate the performance of Policies 1, 2, and 3 for each of the twelve possible flexibility structures with four skills, we randomly generated 50,000 Markovian systems with service rates μ_{ij}, where $i \in \{1, 2\}$ and $j \in \{1, 2, 3\}$, independently drawn from a uniform distribution with range [0.5,2.5] and the buffer sizes B_2, B_3 independently drawn from the discrete uniform distribution with range $\{0, \ldots, 10\}$. For each system and flexibility structure, we use the policy iteration algorithm to find the optimal throughput of the system. Table 3 shows the 95% confidence interval for the throughput of Policies 1, 2, and 3 (determined by solving the balance equations for the Markov chain in each experiment) and the throughput of the optimal policy. The first column shows the flexibility structure under consideration. More specifically, the first set of numbers shows the stations where server 1 is cross-trained at, and the second set of numbers (after the "-" sign) shows the stations where server 2 is cross-trained at. Note that the throughputs of Policies 1, 2, and 3 are the same for the four flexibility structures with one dedicated and one fully flexible server where these policies are identical.

Table 3: Performance of Heuristics for Partially Flexible Understaffed Systems

Flexibility Structure	Policy 1	Policy 2	Policy 3	Optimal Policy
1-123	0.6798 ± 0.0018	0.6798 ± 0.0018	0.6798 ± 0.0018	0.6798 ± 0.0018
123-1	0.6802 ± 0.0018	0.6802 ± 0.0018	0.6802 ± 0.0018	0.6802 ± 0.0018
2-123	0.6698 ± 0.0018	0.6772 ± 0.0018	0.6772 ± 0.0018	0.6780 ± 0.0018
123-2	0.6685 ± 0.0018	0.6760 ± 0.0018	0.6760 ± 0.0018	0.6769 ± 0.0018
3-123	0.6797 ± 0.0018	0.6797 ± 0.0018	0.6797 ± 0.0018	0.6797 ± 0.0018
123-3	0.6813 ± 0.0018	0.6813 ± 0.0018	0.6813 ± 0.0018	0.6813 ± 0.0018
12-13	0.8690 ± 0.0021	0.8721 ± 0.0021	0.8734 ± 0.0021	0.8752 ± 0.0021
13-12	0.8647 ± 0.0021	0.8689 ± 0.0021	0.8701 ± 0.0021	0.8719 ± 0.0021
12-23	0.8716 ± 0.0021	0.8755 ± 0.0021	0.8769 ± 0.0021	0.8801 ± 0.0021
23-12	0.8703 ± 0.0021	0.8742 ± 0.0021	0.8757 ± 0.0021	0.8789 ± 0.0021
13-23	0.8660 ± 0.0021	0.8702 ± 0.0021	0.8726 ± 0.0021	0.8741 ± 0.0021
23-13	0.8671 ± 0.0021	0.8714 ± 0.0021	0.8732 ± 0.0021	0.8755 ± 0.0021

From Table 3, we see that Policy 2 attained more that 99% of the optimal throughput in all the flexibility structures we considered. Moreover, when $D \in \{1, 3\}$, Policy 2 attained the optimal throughput in all the experiments we performed. When $D = 2$, Policy 2 reached 99.95% of the optimal throughput. The reason why the flexibility structures with a dedicated server at either end station perform better compared to the one with a dedicated server at station 2 may be that when the dedicated server is at one of the end stations, the flexible server can focus on making sure the dedicated server is not blocked (if $D = 1$) or not starved (if $D = 3$). By contrast, if the dedicated server is at station 2, the flexible server has to attempt to make sure the dedicated server is neither blocked nor starved.

When both servers have two skills, we observe that Policy 1 performs well and Policy 2 performs even better (it reaches more than 99% of the optimal throughput in all cases). Finally, Policy 3 seems to close the optimality gap for Policy 2 by about 50% when there are two partially flexible servers. Moreover, the flexibility structure with $F = 2$ seems to perform better compared to the ones with $F \in \{1, 3\}$. This is reasonable because training both servers at the middle station allows each server to simultaneously be able to concentrate on one end of the line while being able to help with the operation of the middle station. We also observe that when Policy 1 is employed for systems with two partially flexible servers, the flexibility structure with $F = 3$ performs statistically better than the flexibility structure with $F = 1$. This is consistent with our results about the optimal policy for such systems in Section 4.4, where the optimal policy for the case with $F = 3$ and $B_2 = B_3 = 0$ was shown to be a strict priority policy (as in Policy 1) and the optimal policy for the case with $F = 1$ and $B_2 = B_3 = 0$ was shown to be a threshold policy for some service rates.

We conclude that server assignment policies that are of priority or threshold type are also effective in systems with larger buffers sizes. The optimal policies described in Theorem 4.1 for the cases where $D \in \{1, 3\}$ appear to be optimal for systems with larger buffer sizes as well. For the other cases, the form of the optimal server assignment policy seems complicated (as in Section 4.1), but it is still possible to attain near-optimal throughput with the simple heuristics described in this section.

5.2 Comparison with Full Flexibility and Selection of an Effective Flexibility Structure

In this section, we compare the performance of partially flexible lines with four skills with the optimal performance of the corresponding fully flexible system. We perform 50,000 experiments, as described in Section 5.1. In each experiment, we first use the criteria in Theorem 3.3 to select a flexibility structure (that is known to be optimal for deterministic systems with finite buffers) and use either a heuristic (Policy 3 of Section 5.1) or the optimal server assignment policy to determine the throughput for this flexibility structure. The resulting 95% confidence intervals for the long-run average throughput are shown in the second column of Table 4. Then we determine the throughput of the best heuristic (Policy 3) and optimal policy for each of the twelve flexibility structures with four skills (that are shown in the first column of Table 3), and the structure with the highest throughput is selected. The resulting 95% confidence intervals on the throughput are provided in the third column of Table 4. Finally, the last column of Table 4 provides a 95%

confidence interval on the optimal long-run average throughput of the fully flexible system (we have not considered any heuristic server assignment policy for the fully flexible system).

Table 4: Comparison of the Throughput of Four-Skilled Systems with Six-Skilled Systems

Policy	Theorem 3.3	Best 4-skilled	6-skilled
Best Heuristic	1.0651 ± 0.0018	1.0694 ± 0.0018	—
Optimal	1.0712 ± 0.0018	1.0763 ± 0.0018	1.0865 ± 0.0018

We see from Table 4 that 98.42% of the benefits of full flexibility can be attained with only four skills and our heuristic Policy 3, even with small buffer sizes. When the optimal assignment policy is used with the best four-skilled flexibility structure, we see that the average throughput is 99.06% of that of the fully flexible system. Observe that the optimality gap is caused by the lack of two skills is larger than the optimality gap caused by the use of a heuristic server assignment policy. Moreover, the criteria used for selecting the best flexibility structure when the service times are deterministic also work well for the Markovian system (attaining 98.03% and 98.59% of the throughput of the fully flexible system when Policy 3 and the optimal server assignment policy are employed, respectively).

Table 5 gives the frequency with which each flexibility structure is chosen in the 50,000 sets of service rates using the different criteria. The first column shows the flexibility structure. The second column shows the frequency of selecting each flexibility structure according to the selection rule of Theorem 3.3. The third and fourth columns give the frequency for each flexibility structure when the flexibility structure with the highest throughput is selected and the best heuristic and optimal assignment policies are employed, respectively.

Table 5 shows that the flexibility structures with two partially flexible servers are most of the time superior to the flexibility structures with one dedicated and one fully flexible server. Moreover, there is no big difference in the performance of the systems with a dedicated server at stations 1, 2, or 3, respectively. Similarly, when each server has two skills, the systems that have both servers trained at stations 1, 2, or 3 perform in a similar manner. (For the two approaches that take into account the stochastic nature of the problem, $D = 2$ is worse than $D \in \{1, 3\}$ and $F = 2$ is better that $F \in \{1, 3\}$.)

We conclude that the solution of the allocation LP provides a good heuristic for finding an effective flexibility structure for a tandem line with three stations and two servers. Even though this selection rule does not always identify the best flexibility structure, the average performance of

Table 5: Frequency of Each Flexibility Structure Being the Best in Understaffed Systems

Flexibility Structure	Theorem 3.3	Best Heuristic	Optimal
1-123 or 123-1	2.90%	2.99%	3.06%
2-123 or 123-2	3.00%	2.71%	2.60%
3-123 or 123-3	2.93%	3.01%	2.80%
12-13 or 13-12	30.70%	30.64%	29.62%
12-23 or 23-13	30.26%	31.53%	32.01%
13-23 or 23-13	30.21%	29.12%	29.90%

the flexibility structure it recommends is near-optimal. Furthermore, heuristic server assignment policies for systems with four-skills perform almost as well as optimal server assignment policies, and the frequency with which each flexibility structure is the best is very similar when the heuristic or optimal server assignment policies are employed. In the next section, we will study whether the optimal solution of the allocation LP provides guidance about effective flexibility structures in longer Markovian lines.

5.3 Longer Markovian Lines

In Sections 4.2 and 5.2 we observed that in Markovian systems with three stations and two servers, (i) systems with the same number of skills as the optimal solution of the allocation LP attain near-optimal throughput, and (ii) the optimal flexibility structure of the corresponding deterministic system (found by solving the allocation LP) performs almost as well as the best flexibility structure. In this section, we test these conjectures for longer lines. More specifically, we consider tandem lines with two servers and $N \in \{4, 5\}$ stations. Then Proposition 2 of Andradóttir et al. (2003) shows that the optimal solution of the allocation LP involves $N+1$ skills. We randomly generate the service rates of each server at each station with the same parameters as in Section 4.1. For each set of service rates, we consider three different policies, namely the best policy with N skills (and hence no flexibility), the best policy with the flexibility structure found by solving the allocation LP, and the best policy with $N+1$ skills. In all cases, we find the optimal throughput for each flexibility structure under consideration, and in the case of the best policies with N or $N+1$ skills, we identify the flexibility structure with the stated number of skills yielding the highest optimal throughput. The performance of the three policies is compared with that of the optimal policy for the fully flexible system. We assume $B_j = B$ for all $j \in \{2, \dots, N\}$, where $B \in \{0, 1, 2, 3\}$, and we repeat the experiment for each different value of B.

For systems with two servers and four stations, there are eight possible skills and the optimal solution of the allocation LP involves 5 skills. There are $\binom{8}{5} = 56$ different choices for these five skills, but under our assumptions (i.e., that the service rate of both servers cannot be zero at the same station), it is sufficient to consider 32 different flexibility structures. Similarly, there are $2^4 = 16$ different choices of flexibility structures with four skills, but it suffices to consider 14 choices under our assumptions (since each server has at least one positive service rate). We performed 50,000 experiments for each buffer size, and found the optimal server assignment policy for each of the 14 (32) flexibility structures with 4 (5) skills. Then, we compare the throughputs of the three policies under consideration to the optimal throughput of the corresponding fully flexible system, and the average performance of the three policies (as a percentage of the optimal throughput of the fully flexible system) is shown in columns 2 through 4 of Table 6. Similarly, for the system with two flexible servers and five stations, there are ten possible skills and we know that there exists a six-skilled system that reaches the maximal capacity when the allocation LP is solved. Under our assumptions on the service rates, there are 80 different flexibility structures to consider with six skills, and 30 possibilities with five skills. Because of the prohibitive amount of computational time, the number of experiments are 50,000, 10,000, 5,000, and 1,000, for $B = 0, 1, 2$ and 3, respectively. The results are given in columns 5 through 7 of Table 6.

Table 6: Comparison of Dedicated and Partially Flexible Systems with Fully Flexible Systems

	$N=4$			$N=5$		
Buffer Sizes	Best 4-skilled	Allocation LP	Best 5-skilled	Best 5-skilled	Allocation LP	Best 6-skilled
$B_2 = B_3 = 0$	81.27%	92.17%	93.14%	83.15%	91.22%	92.11%
$B_2 = B_3 = 1$	85.59%	96.21%	96.80%	86.44%	94.20%	94.92%
$B_2 = B_3 = 2$	87.02%	97.73%	98.17%	87.63%	95.07%	95.98%
$B_2 = B_3 = 3$	87.64%	98.25%	98.89%	88.04%	95.81%	96.76%

For both systems we consider, even for buffer sizes as small as zero, the throughput of the best partially flexible system (with $N+1$ skills) is near-optimal compared to the fully flexible system, attaining more than 92% of the optimal throughput in all cases. When $B = 3$, the discrepancy between the performance of partial and fully flexible systems is 1.11% when $N = 4$ and 2.24% when $N = 5$. Hence, we conclude that the allocation LP provides a good guideline for selecting the number of skills in a flexibility structure even for longer Markovian lines with finite buffers. We also observe that the flexibility structure obtained by solving the allocation LP performs very similarly to the best flexibility structure with the same number of skills. Moreover, we see that that adding one more skill to the system with no flexibility closes the optimality gap by at least 63% and sometimes by more than 91% in the systems that we consider. However, the marginal benefit in systems with four and five stations is lower compared to systems with three stations (see Table 2). This is in part because the performance of the systems with no flexibility improves as the number of stations increases (see Tables 2 and 6).

6 Conclusions and Managerial Insights

In this paper, we study understaffed tandem lines with finite buffers. More specifically, we determine throughput-optimal server assignment policies for systems with three stations, two servers possessing four skills in total, and either deterministic service times and arbitrary buffer sizes, or exponential service times and small buffer sizes. Our results for deterministic systems show that it is possible to attain the benefits of full flexibility and infinite buffers with only partial flexibility and zero buffers, and we identify the optimal cross-training strategy for such systems. For Markovian systems, we observe that the optimal assignment policy is of either priority or threshold type. We use the optimal policies for deterministic and small Markovian systems to develop heuristic server assignment policies for Markovian systems with larger buffers, and present empirical results that show that these heuristics perform well. Our numerical experiments also show that the partial flexibility structure that is optimal for a deterministic, infinite-buffered system, together with an effective server assignment policy, achieves near-optimal throughput for the corresponding Markovian system, even for small buffer sizes and longer understaffed lines.

Our research provides the following managerial insights:

- It is possible to attain most of the benefits of full flexibility with partial flexibility.
- An effective flexibility structure that is robust to service time distributions and buffer sizes can be identified by solving a linear program.
- The efficiency loss due to the finite buffer size can be alleviated (and even eliminated in deterministic systems) through the right choice of critical skills.

- Making all servers partially flexible is usually better than making some of them dedicated and some of them fully flexible.
- When there is a dedicated server, the priority of the flexible server should be to prevent the starving or blocking of the dedicated server.
- The priorities of the servers may depend on the buffer sizes (i.e., the best policy for one set of buffer sizes is not necessarily best for another set).

References

H.-S. Ahn and R. Righter. Dynamic load balancing with flexible workers. *Advances in Applied Probability*, 38:621–642, 2006.

H.-S. Ahn, I. Duenyas, and R. Zhang. Optimal scheduling of a 2-stage tandem queue with parallel servers. *Advances in Applied Probability*, 31:1095–1117, 1999.

H.-S. Ahn, I. Duenyas, and M. E. Lewis. The optimal control of a two-stage tandem queueing system with flexible servers. *Probability in the Engineering and Informational Sciences*, 16: 453–469, 2002.

H.-S. Ahn, I. Duenyas, and R. Zhang. Optimal control of a flexible server. *Advances in Applied Probability*, 36:139–170, 2004.

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

S. Andradóttir, H. Ayhan, and D. G. Down. Server assignment policies for maximizing the steady-state throughput of finite queueing systems. *Management Science*, 47:1421–1439, 2001.

S. Andradóttir, H. Ayhan, and D. G. Down. Dynamic server allocation for queueing networks with flexible servers. *Operations Research*, 51:952–968, 2003.

S. Andradóttir, H. Ayhan, and D. G. Down. Compensating for failures with flexible servers. *Operations Research*, 55:753–768, 2007a.

S. Andradóttir, H. Ayhan, and D. G. Down. Dynamic assignment of dedicated and flexible servers in tandem lines. *Probability in the Engineering and Informational Sciences*, 21:497–538, 2007b.

S. Andradóttir, H. Ayhan, and D. G. Down. Maximizing the throughput of tandem lines with failure-prone servers and finite buffers. *Probability in the Engineering and Informational Sciences*, 22:191–211, 2008.

R. Arumugan, M. E. Mayorga, and K. M. Taaffe. Inventory based allocation policies for flexible servers in serial systems. *Annals of Operations Research*, 2008. In print.

J. J. Bartholdi, III and D. D. Eisenstein. A production line that balances itself. *Operations Research*, 44:21–34, 1996a.

J. J. Bartholdi, III and D. D. Eisenstein. Bucket brigade web page, 1996b. URL www.bucketbrigades.com, accessed July 2008.

J. J. Bartholdi, III and S. T. Hackman. Warehouse & Distribution Science. URL http://www2.isye.gatech.edu/ jjb/wh/book/editions/wh-sci-0.87.pdf, accessed July 2008, 2008.

J. J. Bartholdi, III, L. A. Bunimovich, and D. D. Eisenstein. Dynamics of two- and three-worker "bucket brigade" production lines. *Operations Research*, 47:488–491, 1999.

J. J. Bartholdi, III, D. D. Eisenstein, and R. D. Foley. Performance of bucket brigades when work is stochastic. *Operations Research*, 49:710–719, 2001.

S. L. Bell and R. J. Williams. Dynamic scheduling of a system with two parallel servers in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Annals of Applied Probability*, 11:608–649, 2001.

S. L. Bell and R. J. Williams. Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Electronic Journal of Probability*, 10:1044–1115, 2005.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusets, 1997.

I. Duenyas, D. Gupta, and T. L. Olsen. Control of a single server queueing system with setups. *Operations Research*, 46:218–230, 1998.

T. M. Farrar. Optimal use of an extra server in a two station tandem queueing network. *IEEE Transactions on Automatic Control*, 38:1296–1299, 1993.

E. S. Gel, W. J. Hopp, and M. P. VanOyen. Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IIE Transactions*, 34:847–863, 2002.

S. C. Graves and B. T. Tomlin. Process flexibility in supply chains. *Management Science*, 49: 907–919, 2003.

S. Gurumurthi and S. Benjaafar. Modeling and analysis of flexible queueing systems. *Naval Research Logistics*, 51:755–782, 2004.

B. Hajek. Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control*, 29:491–499, 1984.

J. M. Harrison and M. J. López. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems*, 33:339–368, 1999.

W. J. Hopp and M. P. Van Oyen. Agile workforce evaluation: A framework for cross-training and coordination. *IIE Transactions*, 36:919–940, 2004.

W. J. Hopp, E. Tekin, and M. P. Van Oyen. Benefits of skill chaining in serial production lines with cross-trained workers. *Management Science*, 50:83–98, 2004.

S. M. R. Iravani, M. J. M. Posner, and J. M. Buzacott. A two-stage tandem queue attended by a moving server with holding and switching costs. *Queueing Systems*, 26:203–228, 1997.

W. C. Jordan and S. C. Graves. Principles on the benefits of manufacturing flexibility. *Management Science*, 41:577–594, 1995.

D. L. Kaufman, H.-S. Ahn, and M. E. Lewis. On the introduction of an agile, temporary workforce into a tandem queueing system. *Queueing Systems*, 51:135–171, 2005.

H. E. Kırkızlar. *Performance Improvements through Flexible Workforce.* PhD thesis, Georgia Institute of Technology, 2008.

H. E. Kırkızlar, S. Andradóttir, and H. Ayhan. Robustness of effective server assignment policies to service time distributions. Submitted for publication, 2008.

Y. F. Lim and K. K. Yang. Maximizing throughput of bucket brigades on discrete work stations. *Production and Operations Management*, 18:48–59, 2009.

S. A. Lippman. Applying a new device in the optimization of exponential queueing system. *Operations Research*, 23:687–710, 1975.

A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule. *Operations Research*, 52:836–855, 2004.

J. O. McClain, L. J. Thomas, and C. Sox. On-the-fly line balancing with very little WIP. *International Journal of Production Economics*, 27:283–289, 1992.

J. Ostolaza, J. O. McClain, and L. J. Thomas. The use of dynamic (state-dependent) assembly-line balancing to improve throughput. *Journal of Manufacturing and Operations Management*, 3:105–133, 1990.

M. L. Puterman. *Markov Decision Processes*. John Wiley & Sons, New York, NY, 1994.

Z. Rosberg, P. P. Varaiya, and J. C. Walrand. Optimal control of service in tandem queues. *IEEE Transactions on Automatic Control*, 27:600–609, 1982.

M. Sheikhzadeh, S. Benjaafar, and D. Gupta. Machine sharing in manufacturing systems: Total flexibility versus chaining. *The International Journal of Flexible Manufacturing Systems*, 10: 351–378, 1998.

L. Tassiulas and P. P. Bhattacharya. Allocation of interdependent resources for maximal throughput. *Stochastic Models*, 16:27–48, 2000.

E. von Hippel. Task partitioning: An innovation process variable. *Research Policy*, 19:407–418, 1990.

R. B. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing & Service Operations Management*, 7:276–294, 2005.

R. J. Williams. On dynamic scheduling of a parallel server system with complete resource pooling. In *Analysis of Communication Networks: Call Centres, Traffic and Performance*, pages 49–71. Fields Institute Communications Volume 28, American Mathematical Society, 2000.

C.-H. Wu, M. E. Lewis, and M. Veatch. Dynamic allocation of reconfigurable resources in a two-stage tandem queueing system with reliability considerations. *IEEE Transactions on Automatic Control*, 51:309–314, 2006.

C.-H. Wu, D. G. Down, and M. E. Lewis. Heuristics for allocation of reconfigurable resources in a serial line with reliability considerations. *IIE Transactions*, 40:595–611, 2008.

E. Zavadlav, J. O. McClain, and L. J. Thomas. Self-buffering, self-balancing, self-flushing production lines. *Management Science*, 42:1151–1164, 1996.

Appendices

A Proof of Theorem 3.1

We only provide the proof for the case where $D = 1$ (i.e., $\mu_{12} = \mu_{13} = 0$). When $D \in \{2, 3\}$, similar calculations provided in Kırkızlar (2008) show that the policy of the theorem is optimal.

When $D = 1$, the allocation LP takes the simpler form:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{11} + \delta_{21}\mu_{21} \geq \lambda, && (3)\\
& \delta_{22}\mu_{22} \geq \lambda, && (4)\\
& \delta_{23}\mu_{23} \geq \lambda, && (5)\\
& \delta_{21} + \delta_{22} + \delta_{23} \leq 1, \\
& \delta_{2j} \geq 0, \text{ for all } j \in \{1, 2, 3\}.
\end{aligned}$$

Note that our assumptions on the service rates imply that $\mu_{11}, \mu_{22}, \mu_{23} > 0$, and our assumption that server 2 is fully flexible implies that $\mu_{21} > 0$. If $\mu_{11} > \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$, we see that the left-hand side of the constraint (3) is always bigger than the left-hand sides of the constraints (4) and (5), and hence we have $\delta^*_{21} = 0$ in the optimal solution. Then, we find $\delta^*_{22} = \frac{\mu_{23}}{\mu_{22}+\mu_{23}}$ and $\delta^*_{23} = \frac{\mu_{22}}{\mu_{22}+\mu_{23}}$, by solving the equations $\delta^*_{22}\mu_{22} = \delta^*_{23}\mu_{23}$ and $\delta^*_{22} + \delta^*_{23} = 1$. On the other hand, if $\mu_{11} \leq \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$, then we see that in the optimal solution all the constraints (3), (4), and (5) will be tight. Then, we find $\delta^*_{22} = \frac{\mu_{23}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}}$, $\delta^*_{23} = \frac{\mu_{22}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}}$, and $\delta^*_{21} = 1 - \delta^*_{22} - \delta^*_{23}$, by solving the equations $\mu_{11} + \delta^*_{21}\mu_{21} = \delta^*_{22}\mu_{22} = \delta^*_{23}\mu_{23}$ and $\delta^*_{21} + \delta^*_{22} + \delta^*_{23} = 1$. Consequently, the value of λ^* in the optimal solution is as follows:

$$\lambda^* = \begin{cases} \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}} & \text{if } \mu_{11} > \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}, \\ \frac{\mu_{22}\mu_{23}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} & \text{if } \mu_{11} \leq \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}. \end{cases} \tag{6}$$

Now, consider the policy described in the theorem and assume that the system returns to state $s^0 = (s^0_1, s^0_2)$ at time $T > 0$. Note that the remaining service times at all three stations equal one at time T. When $\mu_{11} > \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$ (i.e., $\frac{1}{\mu_{11}} < \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$), server 1 can complete the job at station 1 before server 2 finishes processing a job at stations 2 and 3; hence server 2 does not help server 1. The states of the system and the remaining service requirements for the jobs at each station will be as in Table 7. When $\mu_{11} \leq \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$ (i.e., $\frac{1}{\mu_{11}} \geq \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$), server 2 finishes processing a job at stations 2 and 3, and helps server 1 afterwards. The states of the system and the remaining service requirements for the jobs at each station will be as in Table 8. (In the last column of

Tables 7 and 8, we use the convention that the service requirement for a station is equal to one if this station is starved.) We see that the system regenerates each time it hits the state s^0, that there is one departure from the system during each regenerative cycle, and that the length of the cycle is equal to the reciprocal of equation (6). Hence, we conclude that the policy given in the theorem is optimal. □

Table 7: Sample Path for the Understaffed System with $\mu_{12} = \mu_{13} = 0$, and $\mu_{11} \geq \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$.

Case	Time	State	Remaining Service Requirement
$\frac{1}{\mu_{11}} < \frac{1}{\mu_{22}}$	T	(s_1^0, s_2^0)	$(1, 1, 1)$
	$T + \frac{1}{\mu_{11}}$	$(s_1^0 + 1, s_2^0)$	$(1, 1 - \frac{1}{\mu_{11}}\mu_{22}, 1)$
	$T + \frac{1}{\mu_{22}}$	$(s_1^0, s_2^0 + 1)$	$(1, 1, 1)$
	$T + \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$	(s_1^0, s_2^0)	$(1, 1, 1)$
$\frac{1}{\mu_{11}} > \frac{1}{\mu_{22}}$	T	(s_1^0, s_2^0)	$(1, 1, 1)$
	$T + \frac{1}{\mu_{22}}$	$(s_1^0 - 1, s_2^0 + 1)$	$(1 - \frac{1}{\mu_{22}}\mu_{11}, 1, 1)$
	$T + \frac{1}{\mu_{11}}$	$(s_1^0, s_2^0 + 1)$	$(1, 1, 1 - \left(\frac{1}{\mu_{11}} - \frac{1}{\mu_{22}}\right)\mu_{23})$
	$T + \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$	(s_1^0, s_2^0)	$(1, 1, 1)$
$\frac{1}{\mu_{11}} = \frac{1}{\mu_{22}}$	T	(s_1^0, s_2^0)	$(1, 1, 1)$
	$T + \frac{1}{\mu_{22}}$	$(s_1^0, s_2^0 + 1)$	$(1, 1, 1)$
	$T + \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$	(s_1^0, s_2^0)	$(1, 1, 1)$

Table 8: Sample Path for the Understaffed System with $\mu_{12} = \mu_{13} = 0$, and $\mu_{11} < \frac{\mu_{22}\mu_{23}}{\mu_{22}+\mu_{23}}$.

Time	State	Remaining Service Requirement
T	(s_1^0, s_2^0)	$(1, 1, 1)$
$T + \frac{1}{\mu_{22}}$	$(s_1^0 - 1, s_2^0 + 1)$	$(1 - \frac{1}{\mu_{22}}\mu_{11}, 1, 1)$
$T + \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$	$(s_1^0 - 1, s_2^0)$	$(1 - \left(\frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}\right)\mu_{11}, 1, 1)$
$T + \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}} + \left(\frac{1}{\mu_{11}+\mu_{21}}\right)\left(1 - \mu_{11}\left(\frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}\right)\right)$	(s_1^0, s_2^0)	$(1, 1, 1)$

B Proof of Theorem 3.3

In the interest of space, we only provide the proof of case $\{a\}$ (the proofs of the other cases are are provided in Kırkızlar, 2008). It suffices to show that the optimal value of λ in the allocation LP (2) in the presence of fully flexible servers is equal to the throughput of the system with partially flexible servers. First, we transform this LP to the standard form as follows:

$$\begin{aligned}
\min \quad & -\lambda \\
\text{s.t.} \quad & \lambda - \delta_{1j}\mu_{1j} - \delta_{2j}\mu_{2j} + v_j = 0, \ \text{for } j \in \{1,2,3\}; \\
& \delta_{i1} + \delta_{i2} + \delta_{i3} = 1, \ \text{for } i \in \{1,2\}; \\
& \delta_{ij} \geq 0, \ \text{for all } i \in \{1,2\}, j \in \{1,2,3\}, v_1, v_2, v_3 \geq 0.
\end{aligned} \tag{7}$$

Note that no slack variables are needed in equation (7), because these constraints can be satisfied as equalities without worsening the objective function value. Every feasible basis will have five elements because the LP has five constraints (not including the nonnegativity constraints).

Let D be a basis for the above LP, c_B be the vector of coefficients of the elements of D in the objective function, $\mathbf{B}$ be the coefficients of the elements of D in the constraint matrix, and b be the right-hand side of the constraints. Also, let V be the coefficients of the non-basic variables in the constraint matrix, and c_{NB} be the vector of coefficients of the non-basic variables in the objective function. We let c_B and c_{NB} be row vectors, and b be a column vector. The following conditions guarantee that the basis D is optimal (see, e.g., Theorem 3.1 of Bertsimas and Tsitsiklis, 1997):

$$\mathbf{B}^{-1}b \geq 0, \tag{8}$$

$$c_{NB} - c_B\mathbf{B}^{-1}V \geq 0. \tag{9}$$

Consider the basis $D = \{\lambda, \delta_{11}, \delta_{21}, \delta_{22}, \delta_{23}\}$. With some algebra, we have

$$\mathbf{B}^{-1}b = \begin{bmatrix} \frac{\mu_{22}\mu_{23}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{22}\mu_{23}-\mu_{11}(\mu_{22}+\mu_{23})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{23}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{22}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ 1 \end{bmatrix},$$

$$c_{NB} - c_B\mathbf{B}^{-1}V = \begin{bmatrix} \frac{\mu_{23}(\mu_{11}\mu_{22}-\mu_{12}\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{22}(\mu_{11}\mu_{23}-\mu_{13}\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{22}\mu_{23}}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{21}\mu_{23}}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \\ \frac{\mu_{21}\mu_{22}}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}} \end{bmatrix}.$$

Hence we can conclude that $D = \{\lambda, \delta_{11}, \delta_{21}, \delta_{22}, \delta_{23}\}$ is an optimal basis if conditions $\{1\}$, $\{3\}$, and $\{7\}$ hold. The first element in the matrix $\mathbf{B}^{-1}b$ is the value of λ in the optimal basis, hence $\lambda^* = \frac{\mu_{22}\mu_{23}(\mu_{11}+\mu_{21})}{\mu_{21}\mu_{22}+\mu_{21}\mu_{23}+\mu_{22}\mu_{23}}$ in this case. This result also implies that cross-training server 2 at all stations and server 1 at only station 1 corresponds to the best flexibility structure when case a holds. Furthermore, the policy of Theorem 3.1 attains the maximal capacity in this case, hence it is the optimal server assignment policy. □

C Proof of Proposition 3.1

Consider a set of service rates $\mathcal{R} = \{\mu_{ij} \mid i = 1, 2 \text{ and } j = 1, 2, 3\}$. The elements of $\mathcal{R}$ have to satisfy one of conditions $\{1\}$ and $\{2\}$, one of conditions $\{3\}$ and $\{4\}$, and one of conditions $\{5\}$ and $\{6\}$. First assume that the elements of $\mathcal{R}$ satisfy the conditions $\{1\}$, $\{3\}$, and $\{5\}$. Then, only cases a, f, and i can hold. Note that $\{7\}$ (which is equivalent to $\frac{1}{\mu_{11}} \geq \frac{1}{\mu_{22}} + \frac{1}{\mu_{23}}$) and $\{17\}$ (which is equivalent to $\frac{1}{\mu_{23}} \geq \frac{1}{\mu_{11}} + \frac{1}{\mu_{12}}$) are mutually exclusive, since we assumed that all the service rates are finite. If in addition to $\{1\}$, $\{3\}$, and $\{5\}$, condition $\{7\}$ is satisfied, then conditions $\{8\}$ and $\{17\}$ are not satisfied, and hence only case a holds. If in addition to $\{1\}$, $\{3\}$, and $\{5\}$, condition $\{17\}$ is satisfied, then conditions $\{7\}$ and $\{18\}$ are not satisfied, and hence only case f holds. Finally, if conditions $\{1\}$, $\{3\}$, and $\{5\}$ are satisfied and both of the conditions $\{7\}$ and $\{17\}$ are not satisfied, then conditions $\{8\}$ and $\{18\}$ are satisfied, and hence only case i holds.

Similar arguments show that if conditions $\{1\}$, $\{3\}$, and $\{6\}$ are satisfied, then exactly one of cases a, d, and k holds. If conditions $\{1\}$, $\{4\}$, and $\{6\}$ are satisfied, then exactly one of cases d, e, and h holds. If conditions $\{2\}$, $\{3\}$, and $\{5\}$ are satisfied, then exactly one of cases c, f, and g holds. If conditions $\{2\}$, $\{4\}$, and $\{5\}$ are satisfied, then exactly one of cases b, c, and l holds. If conditions $\{2\}$, $\{4\}$, and $\{6\}$ are satisfied, then exactly one of cases b, e, and j holds.

Finally, note that conditions $\{1\}$, $\{4\}$, and $\{5\}$ cannot hold at the same time because conditions $\{1\}$ and $\{4\}$ together imply that condition $\{6\}$ is true. Similarly, the conditions $\{2\}$, $\{3\}$, and $\{6\}$ cannot hold at the same time because conditions $\{2\}$ and $\{3\}$ together imply that condition $\{5\}$ is true. This concludes the proof. □

D Proof of Theorem 4.1

We only provide the proof for the case where $D = 1$ (i.e., $\mu_{12} = \mu_{13} = 0$). When $D \in \{2, 3\}$, similar calculations provided in Kırkızlar (2008) show that the policy of the theorem is optimal.

When the service times are exponentially distributed, $\{X^\pi(t)\}$ is a continuous-time Markov chain with state space S for all $\pi \in \Pi$, see Section 2. Lemma 2.1 shows that it suffices to consider the policies that are non-idling, and Lemma 2.1 of Kırkızlar et al. (2008) shows that the actions a_{21} and a_{31} are never optimal. Then, the set of allowable actions in state $s \in S$ is

$$A_s = \begin{cases} \{a_{11}\} & \text{for } s = (0,0), \\ \{a_{12}\} & \text{for } s = (B_2+2, 0), \\ \{a_{13}\} & \text{for } s = (B_2+1, B_3+2), \\ \{a_{11}, a_{12}, a_{22}\} & \text{for } s = (i, 0), \text{ where } i \in \{1, \ldots, B_2+1\}, \\ \{a_{11}, a_{13}, a_{33}\} & \text{for } s = (0, j) \text{ or } s = (i, B_3+2), \text{ where} \\ & \quad i \in \{1, \ldots, B_2\} \text{ and } j \in \{1, \ldots, B_3+2\}, \\ \{a_{12}, a_{13}\} & \text{for } s = (B_2+2, j), \text{ where } j \in \{1, \ldots, B_3+1\}, \\ \{a_{11}, a_{12}, a_{13}, a_{22}, a_{33}\} & \text{for } s = (i, j), \text{ where } i \in \{1, \ldots, B_2+1\} \\ & \quad \text{and } j \in \{1, \ldots, B_3+1\}. \end{cases}$$

Note that we used the fact that assigning a server to a station that is blocked or starved is equivalent to idling this server. Furthermore, in the states where more than one station is operating, it is necessary to consider the actions where both servers are assigned to the same station (even if one server is not cross-trained at that station) because Lemma 2.1 distinguishes idling actions from actions that assign a server to a station where (s)he is unable to work. Under our assumptions on the service rates ($\sum_{i=1}^{M} \mu_{ij} > 0$ for $j \in \{1, \ldots, N\}$, $\sum_{j=1}^{N} \mu_{ij} > 0$ for $i \in \{1, \ldots, M\}$, and $\mu_{12} = \mu_{13} = 0$), it is clear that $\mu_{11} > 0$, $\mu_{22} > 0$, and $\mu_{23} > 0$. Furthermore, for all $\pi \in \Pi$, $\{X^\pi(t)\}$ is uniformizable with the uniformization constant $q = \mu_{11} + \mu_{21} + \mu_{22} + \mu_{23}$ (see, Lippman, 1975); let $\{Y^\pi(t)\}$ be the corresponding discrete time Markov chain. Moreover, Andradóttir et al. (2001) shows that our optimization problem (1) is equivalent to the Markov decision problem of maximizing the steady-state departure rate from the third station for $\{Y^\pi(t)\}$. The policy described in the theorem corresponds to an irreducible Markov chain, and consequently we have a communicating Markov decision process. Thus, we can use the policy iteration algorithm for communicating models as described in Section 9.5.1 of Puterman (1994).

Note that $\pi = (d)^\infty$ for every policy π in Π, where d is the corresponding decision rule with $d(s) \in A_s$ for all $s \in S$. Similarly, let P_d be the probability transition matrix corresponding to the policy π, and $r_d(s)$ denote the reward in state s when policy π is employed.

We start the policy iteration algorithm by considering the policy $\pi_0 = (d_0)^\infty$, where

$$d_0(s) = \begin{cases} a_{11} & \text{for } s = (0,0), \\ a_{12} & \text{for } s = (i,j), \text{ where } i \in \{1,\ldots,B_2+2\} \text{ and } j \in \{0,\ldots,B_3+1\}, \\ a_{13} & \text{for } s = (0,j) \text{ or } s = (i,B_3+2), \text{ where } i \in \{1,\ldots,B_2+1\} \\ & \qquad \text{and } j \in \{1,\ldots,B_3+2\}. \end{cases}$$

Then we obtain

$$r_{d_0}(s) = \begin{cases} 0 & \text{for } s = (0,0) \text{ or } s = (i,j), \text{ where } i \in \{1,\ldots,B_2+2\} \\ & \qquad \text{and } j \in \{0,\ldots,B_3+1\}, \\ \mu_{23} & \text{for } s = (0,j) \text{ or } s = (i,B_3+2), \text{ where } i \in \{1,\ldots,B_2+1\} \\ & \qquad \text{and } j \in \{1,\ldots,B_3+2\}, \end{cases}$$

and

$$P_{d_0}(s,s') = \begin{cases} \frac{\mu_{11}+\mu_{21}}{q} & \text{for } s = (0,0) \text{ and } s' = (1,0); \\ \frac{\mu_{22}+\mu_{23}}{q} & \text{for } s = s' = (0,0); \\ \frac{\mu_{11}}{q} & \text{for } s = (i,j),\ s' = (i+1,j), \text{ where } i \in \{1,\ldots,B_2+1\} \\ & \qquad \text{and } j \in \{0,\ldots,B_3+1\}; \\ \frac{\mu_{22}}{q} & \text{for } s = (i,j),\ s' = (i-1,j+1), \text{ where } i \in \{1,\ldots,B_2+2\} \\ & \qquad \text{and } j \in \{0,\ldots,B_3+1\}; \\ \frac{\mu_{21}+\mu_{23}}{q} & \text{for } s = s' = (i,j), \text{ where } i \in \{1,\ldots,B_2+1\} \\ & \qquad \text{and } j \in \{0,\ldots,B_3+1\}; \\ \frac{\mu_{11}+\mu_{21}+\mu_{23}}{q} & \text{for } s = s' = (i,j), \text{ where } i = B_2+2 \\ & \qquad \text{and } j \in \{0,\ldots,B_2+1\}; \\ \frac{\mu_{11}}{q} & \text{for either } s = (0,j) \text{ and } s' = (1,j) \text{ or } s = (i,B_3+2) \text{ and } s' = (i+1,B_3+2), \\ & \qquad \text{where } i \in \{1,\ldots,B_2\} \text{ and } j \in \{1,\ldots,B_3+2\}; \\ \frac{\mu_{23}}{q} & \text{for either } s = (0,j) \text{ and } s' = (0,j-1) \text{ or } s = (i,B_3+2) \text{ and } s' = (i,B_3+1), \\ & \qquad \text{where } i \in \{1,\ldots,B_2+1\} \text{ and } j \in \{1,\ldots,B_3+2\}; \\ \frac{\mu_{21}+\mu_{22}}{q} & \text{for } s = s' = (0,j) \text{ or } s = s' = (i,B_3+2), \text{ where } i \in \{1,\ldots,B_2\} \\ & \qquad \text{and } j \in \{1,\ldots,B_3+2\}; \\ \frac{\mu_{11}+\mu_{21}+\mu_{22}}{q} & \text{for } s = s' = (B_2+1,B_2+2). \end{cases}$$

For all $s, s' \in S$ and $a \in A_s$, we use $r(s,a)$ to denote the immediate reward in state s when action a is taken and $p(s'|s,a)$ to denote the one-step probability of going from state s to state s'

when action a is chosen in state s. Since $\{Y^{\pi_0}(t)\}$ is an irreducible Markov chain, we can solve the following set of equations to find a scalar g_0 and a vector h_0, letting $h_0(0,0) = 0$,

$$r_{d_0} - g_0 e + (P_{d_0} - I)h_0 = 0,$$

where e is the column vector of ones and I is the identity matrix. Then, we compute $d(s)$, where

$$d(s) \in \arg\max_{a \in A_s} \Big\{ r(s,a) + \sum_{s' \in S} p(s'|s,a) h_0(s') \Big\}, \quad \forall s \in S,$$

and set $d(s) = d_0(s)$ whenever possible. If one can show $d(s) = d_0(s)$ for all $s \in S$, then the policy $\pi_0 = (d_0)^\infty$ is optimal according to Theorem 9.5.1 of Puterman (1994). Consequently, for all $s \in A_s$ and $a \in A_s$, we want to show that the following inequality holds:

$$\Delta(s,a) = \Big(r(s, d_0(s)) + \sum_{s' \in S} p(s'|s, d_0(s)) h_0(s') \Big) - r(s,a) - \sum_{s' \in S} p(s'|s,a) h_0(s') \geq 0.$$

In the calculations below, $\xi_k(s, B_2, B_3)$ for $k \in \{1, \dots, 11\}$ are nonnegative constants (that depend on the service rates, the state $s \in S$, and the buffer sizes), and $\xi(B_2, B_3)$ is a nonnegative constant (that depends on the service rates and the buffer sizes); they are provided in Kırkızlar (2008). We assume that $B_2, B_3 \leq 1$ in the following calculations. For $s \in \{(0,0), (B_2+2, 0), (B_2+1, B_3+2)\}$, the action $d_0(s)$ is optimal because there is only one action in A_s.

First, consider the state $s = (i, 0)$, where $i \in \{1, \dots, B_2+1\}$, and recall that $d_0(s) = a_{12}$. With some algebra we have

$$\Delta(s, a_{11}) = \frac{\xi_1(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0, \quad \Delta(s, a_{22}) = \frac{\xi_2(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0.$$

Now let $s = (0, j)$, where $j \in \{1, \dots, B_3+2\}$, and recall that $d_0(s) = a_{13}$. Then, we have

$$\Delta(s, a_{11}) = \frac{\xi_3(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0, \quad \Delta(s, a_{33}) = \frac{\xi_4(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0.$$

Similarly, let $s = (i, B_3+2)$, where $i \in \{1, \dots, B_2\}$, and recall that $d_0(s) = a_{13}$. We have

$$\Delta(s, a_{11}) = \frac{\xi_5(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0, \quad \Delta(s, a_{33}) = \frac{\xi_6(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0.$$

For $s = (i, j)$, where $i \in \{1, \dots, B_2+1\}$ and $j \in \{1, \dots, B_3+1\}$, recall that $d_0(s) = a_{12}$. Some algebra shows that

$$\Delta(s, a_{11}) = \frac{\xi_7(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0, \quad \Delta(s, a_{13}) = \frac{\xi_{10}(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0,$$
$$\Delta(s, a_{22}) = \frac{\xi_8(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0, \quad \Delta(s, a_{33}) = \frac{\xi_9(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0.$$

Finally, consider $s = (B_2+2, j)$, where $j \in \{1, \dots, B_3+1\}$, and recall that $d_0(s) = a_{12}$. Some algebra shows that

$$\Delta(s, a_{13}) = \frac{\xi_{11}(s, B_2, B_3)}{\xi(B_2, B_3)} \geq 0. \ \square$$