Queueing Syst
DOI 10.1007/s11134-017-9543-0

Pooling in tandem queueing networks with non-collaborative servers

Nilay Tanık Argon[1] · **Sigrún Andradóttir**[2]

Received: 22 September 2016 / Revised: 26 June 2017

Abstract This paper considers pooling several adjacent stations in a tandem network of single-server stations with finite buffers. When stations are pooled, we assume that the tasks at those stations are pooled but the servers are not. More specifically, each server at the pooled station picks a job from the incoming buffer of the pooled station and conducts all tasks required for that job at the pooled station before that job is placed in the outgoing buffer. For such a system, we provide sufficient conditions on the buffer capacities and service times under which pooling increases the system throughput by means of sample-path comparisons. Our numerical results suggest that pooling in a tandem line generally improves the system throughput—substantially in many cases. Finally, our analytical and numerical results suggest that pooling servers in addition to tasks results in even larger throughput when service rates are additive and the two systems have the same total number of storage spaces.

Keywords Tandem queues · Finite buffers · Production blocking · Throughput · Work-in-process inventory (WIP) · Sample-path analysis · Stochastic orders

Mathematics Subject Classification 90B22 · 60K25

✉ Nilay Tanık Argon
nilay@unc.edu

1 Department of Statistics and Operations Research, University of North Carolina, Chapel Hill, NC 27599-3260, USA

2 H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

1 Introduction

Tandem queueing networks have long been employed as useful models in the design and control of several manufacturing and communications systems. In this paper, we consider such a queueing network where jobs flow through a series of multiple stations each having a single server. Jobs waiting for service at a station queue up in the input buffer of a station which can have limited capacity. This means that a station can be blocked if the input buffer of the downstream station is full. For such a queueing network, we consider *pooling* two or more adjacent stations into a single station with the objective of increasing the long-run average system throughput.

More specifically, we consider a situation where pooling two or more stations results in a single station where the servers of the pooled stations work in parallel on different jobs. Each server takes a job from the input queue and completes the entire service of this job at the pooled station (which consists of the tasks performed at stations that were pooled) without any collaboration with other servers before starting service of another job. Thus, pooling is feasible if the servers at the stations to be pooled are flexible to work at all the pooled stations. Because of the parallel working structure of servers at the pooled station, we refer to this type of pooling as *parallel pooling*. Our main goal in this paper is to study the departure process and throughput of a tandem line in which a group of stations are parallel pooled, and to obtain insights into when such a pooling would be beneficial.

The main work on parallel pooling (see, for example, Smith and Whitt [22], Calabrese [9], Section 8.4.1 of Buzacott and Shanthikumar [8], Benjaafar [6], and Harel [12]) considers resource sharing in unconnected Markovian queuing systems. One conclusion is that pooling parallel queues while keeping the identities of servers is in general beneficial in terms of throughput and congestion measures when all jobs have the same service time distribution. For example, it is well-known that an M/M/m queue with arrival rate $m\lambda$ and service rate μ for each server yields a shorter long-run average waiting time than m parallel M/M/1 queues, each having an arrival rate of λ and service rate μ. However, when parallel queueing systems that serve jobs with different service time distributions are pooled, parallel pooling may degrade the performance, as shown in several studies; see, for example, Smith and Whitt [22] and Benjaafar [6]. Tekin et al. [24] later provided conditions under which parallel pooling of systems with different service time distributions is beneficial by using approximations. For example, they showed that if the mean service times of all jobs are similar, then pooling systems with the highest coefficient of variation for the service times yields the highest reduction in the average delay.

Parallel pooling in queueing networks with identical servers has been also studied before by Buzacott [7], Van Oyen et al. [25], and Mandelbaum and Reiman [17]. Buzacott [7] compares a series system with single-server stations and arrivals at the first station and a system of parallel stations with servers performing all tasks of the series system. The performance measure of interest is the long-run average number of jobs in the system. Assuming that the tasks in the series system are balanced in terms of mean processing times and their coefficients of variation, Buzacott [7] uses multiple approximate formulae (under heavy, medium, and light traffic) to show that the parallel system is better than the tandem line if the jobs in the parallel system are assigned to each server cyclically. However, the author also shows that the opposite is true under heavy traffic if the arriving jobs are assigned to the parallel stations randomly and the service time variability is sufficiently low. Van Oyen et al. [25] consider both parallel pooling and *cooperative pooling* (where all servers are pooled into a single team) in a tandem network, and show that the throughput remains the same under both pooling structures when all stations in the network are pooled. They also provide numerical examples that support the claim that parallel pooling of all stations in a tandem line is an effective policy if the goal is to minimize the mean sojourn time. Finally, under the assumptions of light or heavy traffic, Mandelbaum and Reiman [17] compare parallel and cooperative pooling structures when all stations in a queueing network are pooled. They point out that parallel pooling is always worse than cooperative pooling in terms of the mean sojourn time of each job in the system, even if their steady-state throughputs are the same. Mandelbaum and Reiman [17] also conclude that the difference between the mean steady-state sojourn times of these two pooled systems is maximal in light traffic, and it diminishes as the traffic becomes heavy.

Note that in all prior work on parallel pooling in queueing networks, it is assumed that all servers are identical and all stations in the network are pooled. Moreover, Buzacott [7] and Mandelbaum and Reiman [17] assume that the buffers in the original system are infinite. In this study, we relax these three assumptions and identify sufficient conditions under which parallel pooling of a subset of stations in a tandem line with finite-capacity queues and possibly nonidentical servers will improve the departure process.

Finally, we should note that there is a substantial literature on cooperative pooling in queueing networks. We here mention some of the most relevant work in the area and refer the interested reader to Andradóttir et al. [3] and references therein. Most of the literature on cooperative pooling focuses on dynamic assignment of servers, i.e., situations where servers are not permanently pooled but rather can be dynamically assigned to stations where they can cooperate on the same job, as in Andradóttir et al. [3]. On the other hand, there are a few articles where the decision is about permanently pooling servers into a team. This includes Buzacott [7], Mandelbaum and Reiman [17], and Van Oyen et al. [25], which we mentioned earlier, and Argon and Andradóttir [4]. Argon and Andradóttir [4] consider cooperative pooling of a subset of adjacent stations in a tandem line and study the benefits of such a pooling on the departure process, throughput, work-in-process inventory, and holding costs. The main finding is that pooling a subset of stations in general yields a better outcome, especially when the bottleneck station is pooled, but one needs to be careful about the size and allocation of buffers in the pooled system to realize such a benefit.

It is no surprise that cooperative pooling has been studied more extensively than parallel pooling, as it is generally much easier to analyze models with a single server. However, parallel pooling is a more easily justified pooling mechanism in many applications. For example, in several service systems, such as call centers, pooling many servers into one is undesirable if not impossible. On the other hand, parallel pooling requires that there are enough tools, equipment, and space that multiple jobs can be processed at the same time. Some applications that would satisfy this requirement are office/desk jobs such as code developing and architectural design, service systems such as call centers, and manufacturing processes requiring inexpensive tools and equipment such as textile manufacturing. In these applications, instead of each task being done by a different worker, under parallel pooling multiple tasks of each project/job will be "owned" by a single worker who has access to ample equipment such as computers, phone lines, and sewing machines. For applications where both types of pooling are allowable, it is interesting to compare the effects of these two pooling structures on different performance measures. In this paper, we will use analytical and numerical results to provide insights into this comparison in a tandem line.

The outline of this paper is as follows. In Sect. 2, we analyze the effects of parallel pooling on the departure time of each job from each station in a tandem network and on the steady-state throughput of the system. In Sect. 3, we study the effects of parallel pooling on other performance measures (besides departure times and throughput), namely, the work-in-process inventory, sojourn times, and holding costs. In Sect. 4, we provide a brief comparison of lines with parallel servers and cooperative servers. In Sect. 5, we use numerical results to quantify the potential benefits of parallel pooling and to obtain a better understanding of when pooling with parallel servers will be beneficial in tandem lines with finite buffers. Finally, in Sect. 6, we provide our concluding remarks and discuss some insights that can be drawn from this study. The Appendix provides proofs of our analytical results.

2 Problem formulation and main results

Consider a queueing network of $N \geq 2$ stations in tandem numbered $1, \ldots, N$, where each station $j \in \{1, \ldots, N\}$ has a single server (referred to as server j) and jobs are served in the order that they arrive (i.e., according to the first-come-first-served, FCFS, queueing discipline). We assume that there are $0 \leq b_j \leq \infty$ buffers in front of station $j \in \{2, \ldots, N\}$, an unlimited supply of jobs in front of the first station ($b_1 = \infty$), and an infinite capacity buffer space following the last station ($b_{N+1} = \infty$). Consequently, if all buffers in front of station $j \in \{2, \ldots, N\}$ are full when station $j - 1$ completes a job, then we assume that this job remains at station $j - 1$ until one job at station j is moved to station $j + 1$ or leaves the system (if $j = N$). This type of blocking is usually called *production blocking*. Because we assume that the output buffer space for station N is unlimited, station N will never be blocked.

In the system under consideration, there are at least two adjacent stations whose servers are flexible such that they can work at both of these stations. We let $\mu_{\ell,j} \geq 0$ denote the rate at which server ℓ processes jobs at station j, for $\ell, j \in \{1, \ldots, N\}$. Without loss of generality, we assume that $\mu_{j,j} > 0$ for all $j \in \{1, \ldots, N\}$. The servers are said to be identical if $\mu_{\ell,j} = \mu_{k,j}$ for all $j, k, \ell \in \{1, \ldots, N\}$. We also let $X_j(i)$ be the service time of job $i \geq 1$ at station $j \in \{1, \ldots, N\}$. We call $\mu_{j,j} X_j(i)$ the service requirement of job $i \geq 1$ at station $j \in \{1, \ldots, N\}$.

Now, consider an alternative tandem line, where stations $K, \ldots, M$, for $K \in \{1, \ldots, N - 1\}$ and $M \in \{K + 1, \ldots, N\}$, are pooled to obtain a single station at which servers $K, \ldots, M$ work in parallel. Jobs form a single queue in front of this pooled station and are allocated from this queue to a server only when the server would otherwise be idle. We let $P^{[K,M]}$ and $Q^{[K,M]}$ denote the number of buffers before and after the pooled station, respectively, and assume that the buffer sizes before stations

$2,\ldots,K-1$ and $M+2,\ldots,N$ are kept intact after pooling. We also assume that jobs are served according to the FCFS queueing discipline. Finally, we assume that the blocked jobs at the pooled station are released to station $M+1$ in the order that they became blocked. Hence, the ith service completion and the ith departure from the pooled station are realized by the same job, for $i \geq 1$. In the remainder of this section, we provide sufficient conditions under which such a pooling structure will improve the departure process and throughput of the tandem line under consideration.

Let $X_\ell^{[K,M]}(i)$ be the service time of the ith entering job at the pooled station, for $i \geq 1$, when server $\ell \in \{K,\ldots,M\}$ works on that job. (If it is not known which server is working on the ith entering job at the pooled station, then we suppress the subscript ℓ in $X_\ell^{[K,M]}(i)$.) Although we do not assume it in general, in some results we use the following reasonable model for the service times at the pooled station, which is stated as Assumption 1.

Assumption 1 Assuming that $\mu_{\ell,j} > 0$ for all $\ell, j \in \{K,\ldots,M\}$, the service time of job $i \geq 1$ at the pooled station served by server $\ell \in \{K,\ldots,M\}$ is given by

$$X_\ell^{[K,M]}(i) = \sum_{j=K}^{M} \frac{\mu_{j,j}}{\mu_{\ell,j}} X_j(i). \tag{1}$$

In Assumption 1, $\mu_{j,j}X_j(i)$ represents the service requirement for job i at station j. Hence, when it is divided by $\mu_{\ell,j}$, it gives the service time for job i at station j when processed by server ℓ. Such a scaling of service times for different servers is commonly used in models of flow lines with cross-trained servers; see, for example, [1,5,15].

We let $X_j^{[K,M]}(i)$ denote the service time of the ith entering job at station j, in the pooled system for $j \in \{1,\ldots,K-1,M+1,\ldots,N\}$ and $i \geq 1$. We also let $D_j^{[K,M]}(i)$ be the time of the ith departure from station $j \in \{1,\ldots,K-1,M,\ldots,N\}$, for $i \geq 1$, in the pooled system (we arbitrarily refer to the pooled station as station M). Similarly, we let $D_j(i)$ denote the time of the ith departure from station j in the original line, where $j \in \{1,\ldots,N\}$ and $i \geq 1$. Finally, in order to provide recursive expressions for the departure times from the pooled station, for $j = 1,\ldots,n$ and $n \geq 1$, we define $\Phi_j^{(n)}\{a_1,a_2,\ldots,a_n\}$ to be a function from $\mathbb{R}^n$ to $\mathbb{R}$ that returns the jth largest element in the sequence $\{a_1,a_2,\ldots,a_n\}$ so that $\Phi_1^{(n)}\{a_1,a_2,\ldots,a_n\} \geq \Phi_2^{(n)}\{a_1,a_2,\ldots,a_n\} \geq \cdots \geq \Phi_n^{(n)}\{a_1,a_2,\ldots,a_n\}$.

We next give recursive formulae that the departure times $D_j^{[K,M]}(i)$ must satisfy under the initial condition that all buffers are empty and all servers are idle. For convenience, we assume that $D_j^{[K,M]}(i) = 0$ for $i \leq 0$ or $j \notin \{1,\ldots,K-1,M,\ldots,N\}$. Since there is a single server at stations that are not pooled, for $j \in \{1,\ldots,K-2,M+1,\ldots,N\}$ and $i \geq 1$ we have

$$\begin{aligned} D_j^{[K,M]}(i) = \max\Big\{&D_{j-1}^{[K,M]}(i) + X_j^{[K,M]}(i),\, D_j^{[K,M]}(i-1) + X_j^{[K,M]}(i),\\ &D_{j+1}^{[K,M]}\left(i - b_{j+1} - 1\right)\Big\}. \end{aligned} \tag{2}$$

(Similar dynamic recursions for tandem lines with finite buffers are used by many others such as Argon and Andradóttir [4], Shanthikumar and Yao [21], and references therein.) Moreover, since the pooled station has $P^{[K,M]} + M - K + 1$ storage spaces, including the input buffer and the servers, we have

$$D_{K-1}^{[K,M]}(i) = \max\Big\{D_{K-2}^{[K,M]}(i) + X_{K-1}^{[K,M]}(i), D_{K-1}^{[K,M]}(i-1) + X_{K-1}^{[K,M]}(i),\\ D_M^{[K,M]}\Big(i - P^{[K,M]} - M + K - 1\Big)\Big\}. \tag{3}$$

We next derive a recursive formula for the departure times from the pooled station. For this purpose, we first obtain an expression for the ith service completion time at the pooled station. When the $(i-1)$th departure from the pooled station takes place, then one of the servers can start serving the $(i+M-K)$th job that enters the pooled station, for $i \geq 1$. Hence, the service completion time of the $(i+M-K)$th job that enters the pooled station is given by

$$\max\Big\{D_{K-1}^{[K,M]}(i+M-K), D_M^{[K,M]}(i-1)\Big\} + X^{[K,M]}(i+M-K), \tag{4}$$

for all $i \geq 1$. On the other hand, note that the ith service completion at the pooled station is realized either by the $(i+M-K)$th job that enters the pooled station or by the jobs that enter the pooled station before the $(i+M-K)$th job, but have not yet completed their service requirements at the pooled station at the time of the $(i-1)$th service completion at the pooled station. For $j = 1, \dots, M-K$, let $A_j(1)$ denote the jth largest service completion time at the pooled station among the first $M-K$ jobs that entered the pooled station and let $A_j(i)$, for $i \geq 2$, denote the jth largest service completion time at the pooled station among those $M-K$ jobs that entered the pooled station before the $(i+M-K)$th entering job and have not yet left the pooled station at the time of the $(i-1)$th departure from the pooled station. Hence, the ith service completion from the pooled station is equal to the minimum of $A_{M-K}(i)$ and the service completion time of the $(i+M-K)$th job entering the pooled station. Then, using Eq. (4) and the fact that the ith departure from the pooled station may take place only after departure $i - Q^{[K,M]} - 1$ from station $M+1$ takes place, gives

$$D_M^{[K,M]}(i) = \max\Big\{\min\Big\{\max\Big\{D_{K-1}^{[K,M]}(i+M-K), D_M^{[K,M]}(i-1)\Big\}\\ + X^{[K,M]}(i+M-K), A_{M-K}(i)\Big\}, D_{M+1}^{[K,M]}\Big(i - Q^{[K,M]} - 1\Big)\Big\}. \tag{5}$$

Moreover, for all $j = 1, \dots, M-K$ and $i \geq 1$, we have $A_j(1) = \Phi_j^{(M-K)}\{D_{K-1}^{[K,M]}(m) + X^{[K,M]}(m) : m = 1, \dots, M-K\}$ and

$$A_j(i+1) = \Phi_j^{(M-K+1)}\Big\{\max\Big\{D_{K-1}^{[K,M]}(i+M-K), D_M^{[K,M]}(i-1)\Big\}\\ + X^{[K,M]}(i+M-K), A_1(i), \dots, A_{M-K}(i)\Big\}. \tag{6}$$

Similar recursive formulae for a tandem line with two stations and no buffers in which the first station has a single server and the last station has multiple servers are given in Yamazaki et al. [26]. However, we are not aware of any other work that provides expressions for departure times in a tandem line with parallel servers at a station in this generality.

We use these recursive expressions to prove Proposition 1, which provides a set of conditions on the service times and buffers in the pooled system such that the departures from the pooled system are no later than those from the original (unpooled) line in the sense of sample paths.

Proposition 1 *For* $1 \leq K \leq M \leq N$*, if*

(i) $X_j^{[K,M]}(i) \leq X_j(i)$ *for all* $j \in \{1, \ldots, K-1, M+1, \ldots, N\}$ *and* $i \geq 1$*;*
(ii) $X^{[K,M]}(i) \leq \sum_{k=K}^{M} X_k(i)$ *for all* $i \geq 1$*; and*
(iii) $b_k = 0$ *for* $k \in \{K+1, \ldots, M\}$*,* $P^{[K,M]} \geq b_K$*, and* $Q^{[K,M]} \geq b_{M+1}$*;*

then we have that $D_j^{[K,M]}(i) \leq D_j(i)$ *for* $j \in \{1, \ldots, K-1, M, \ldots, N\}$ *and* $i \geq 1$.

Proposition 1 implies that parallel pooling will result in smaller departure times from the system if (i) service times at stations that are not pooled do not increase by pooling; (ii) the pooled service time of a job at the pooled station is no larger than the total service time of that job at stations $K, \ldots, M$ in the original system, irrespective of which server processes the job at the pooled station; (iii) there are zero buffers between the pooled stations in the original system and the buffers around the pooled station in the pooled system are no smaller than the corresponding buffers in the original line. Defining the throughput of the pooled system by $T^{[K,M]} = \liminf_{i\to\infty}\{i/D_N^{[K,M]}(i)\}$ and that of the original system by $T = \liminf_{i\to\infty}\{i/D_N(i)\}$, Proposition 1 implies that $T^{[K,M]} \geq T$ if conditions (i), (ii), and (iii) are satisfied and the limits exist. (For conditions that guarantee that these limits exist almost surely, see, for example, Proposition 4.8.2 in Glasserman and Yao [10].)

Conditions (i) and (ii) of Proposition 1 are reasonable because they require pooling not to increase service times at each station. Also, under Assumption 1, condition (ii) will hold if $\mu_{j,j} \leq \mu_{\ell,j}$ for all $j, \ell \in \{K, \ldots, M\}$, i.e., if either the servers are identical or the assignment of servers to stations in the original system was poorly done. One would also expect that the result of Proposition 1 may not hold unless the buffers around the pooled station is at least as large as the corresponding buffers in the original line. However, it is harder to justify the condition that the buffers between the pooled stations are zero for pooling to be beneficial. We next provide an example that demonstrates that if this condition does not hold, then the result may fail.

Example 1 Suppose that we pool both stations in a tandem line with two stations and $b_2 \geq 1$. Suppose also that the service times at the pooled station are given by $X_\ell^{[1,2]}(i) = X_1(i) + X_2(i)$ for $\ell = 1, 2$ and $i \geq 1$. Thus, this example satisfies all conditions of Proposition 1 except for the condition that $b_k = 0$ for $k \in \{K+1, \ldots, M\}$. Now, consider a sample path under which the service times for the first four jobs that enter the original system are given by $(X_1(1), X_1(2), X_1(3), X_1(4)) = (1, 5, 10, 5)$ and $(X_2(1), X_2(2), X_2(3), X_2(4)) = (10, 5, 10, 15)$ minutes. Then, we obtain that $D_2(3) = 26$ minutes and $D_2^{[1,2]}(3) = 30$ minutes, i.e., the timing of the third departure from the system is delayed by pooling.

Although the above example demonstrates that the condition that there are zero buffers between the pooled stations in the original system is needed for the result to hold in the sample-path sense, it is not necessarily needed to achieve improvements by pooling in some weaker sense (such as in terms of the long-run average throughput). Indeed, in our numerical experiments presented in Sect. 5, we observe that parallel pooling improves system throughput in most scenarios including those with positive buffers between the pooled stations.

We next provide two results that guarantee an improvement by pooling in a weaker sense than the sample-path sense considered in Proposition 1. We first define the usual stochastic order between two (discrete-time) stochastic processes. Let $\mathcal{Y} = \{\mathbf{Y}(i)\}_{i\geq 1}$ and $\mathcal{Z} = \{\mathbf{Z}(i)\}_{i\geq 1}$ be stochastic processes with state space $\mathbb{R}^d$, where $d \in \mathbb{N}$. Then, $\mathcal{Y}$ is smaller than $\mathcal{Z}$ in the usual stochastic ordering sense ($\mathcal{Y} \leq_{st} \mathcal{Z}$) if and only if $E[f(\mathcal{Y})] \leq E[f(\mathcal{Z})]$ for every non-decreasing functional $f : \mathbb{R}^\infty \to \mathbb{R}$ provided the expectations exist. (A functional $f : \mathbb{R}^\infty \to \mathbb{R}$ is non-decreasing if $f(\{y_1, y_2, \ldots\}) \leq f(\{z_1, z_2, \ldots\})$ whenever $y_i \leq z_i$ for all $i \geq 1$. A functional $\phi : \mathbb{R}^\infty \to \mathbb{R}^\infty$ is non-decreasing if every component of ϕ is non-decreasing.) For more information on the usual stochastic order for stochastic processes, see, for example, Section 6.B.7 in Shaked and Shanthikumar [20].

To simplify our notation, for any vector $\mathbf{Z}(i) = (Z_1(i), \ldots, Z_n(i))$, where $i, n \geq 1$, we define a sub-vector $\mathbf{Z}_{k,\ell}(i) = (Z_k(i), \ldots, Z_\ell(i))$ for $1 \leq k \leq \ell \leq n$. We also define

$$\begin{aligned}
\mathbf{D}(i) &= (D_1(i), \ldots, D_{K-1}(i), D_M(i), D_{M+1}(i), \ldots, D_N(i)),\\
\mathbf{X}(i) &= (X_1(i), \ldots, X_N(i)),\\
\mathbf{D}^{[K,M]}(i) &= (D_1^{[K,M]}(i), \ldots, D_{K-1}^{[K,M]}(i), D_M^{[K,M]}(i), D_{M+1}^{[K,M]}(i), \ldots, D_N^{[K,M]}(i)), \text{ and}\\
\mathbf{X}^{[K,M]}(i) &= (X_1^{[K,M]}(i), \ldots, X_{K-1}^{[K,M]}(i), X^{[K,M]}(i), X_{M+1}^{[K,M]}(i), \ldots, X_N^{[K,M]}(i)), \text{ for all } i \geq 1.
\end{aligned}$$

Proposition 2 *For $1 \leq K \leq M \leq N$, if condition (iii) of Proposition 1 holds and*

$$\left\{\mathbf{X}^{[K,M]}(i)\right\}_{i\geq 1} \leq_{st} \left\{\mathbf{X}_{1,K-1}(i), \sum_{k=K}^{M} X_k(i), \mathbf{X}_{M+1,N}(i)\right\}_{i\geq 1}, \tag{7}$$

then we have that $\left\{\mathbf{D}^{[K,M]}(i)\right\}_{i\geq 1} \leq_{st} \{\mathbf{D}(i)\}_{i\geq 1}$.

Proposition 2 replaces the conditions on service times of Proposition 1 (in particular conditions (i) and (ii)) by the weaker condition (7) at the cost of obtaining an improvement in departure times in the sense of usual stochastic orders. As a stochastic improvement in departure times implies an improvement in the long-run average throughput, the weaker conditions of Proposition 2 are sufficient to guarantee an increase in system throughput by parallel pooling. Note that (7) holds as a stochastic equality when the servers are identical and pooling does not affect task completion times; hence Proposition 2 guarantees improved throughput as long as condition (iii)

of Proposition 1 also holds. We next provide another set of conditions under which parallel pooling of all stations in a tandem line increases the system throughput. We first state one of the main assumptions of this result.

Assumption 2 For $\ell, j \in \{K, \ldots, M\}$, the service rates satisfy the following product form:

$$\mu_{\ell,j} = \theta_\ell \eta_j,$$

where $\theta_\ell \in [0, \infty)$ and $\eta_j \in [0, \infty)$ are constants that depend only on server ℓ and station j, respectively.

Assumption 2 means that the rate of a server working on a job at a station is separable into two components: a component θ_ℓ that quantifies the speed of server ℓ and another component η_j that quantifies the intrinsic difficulty of the task at station j. Hence, this assumption implies that a "fast" server is fast at every station and a "difficult" task is difficult for all servers. In particular, a larger θ_ℓ represents a faster server, whereas a larger η_j represents an easier task. Note that Assumption 2 generalizes the assumption that the service rates depend only on the servers or on the tasks. Several earlier works on queueing systems with flexible servers employed this assumption or special cases thereof; see, for example, [2,5].

Proposition 3 *Suppose that* $\{\mathbf{X}(i)\}_{i\geq 1}$ *is a sequence of independent and identically distributed (i.i.d.) random vectors with* $\mathrm{E}[X_j(i)] < \infty$ *for all* $j \in \{1, \ldots, N\}$ *and* $i \geq 1$. *Then, we have* $T \leq T^{[1,N]}$ *under Assumptions 1 and 2.*

Proposition 3 states that complete pooling (i.e., pooling all stations in a line) increases the throughput under reasonable conditions on the service times and server capabilities. A result similar to Proposition 3 is proved by Buzacott [7] but under the assumption of identical servers and infinite buffers, and later by Van Oyen et al. [25] for identical servers. Proposition 3 also leads to a useful corollary that provides a set of conditions under which partial pooling (i.e., pooling only a subset of stations) increases the system throughput.

Corollary 1 *Suppose that* $\{\mathbf{X}(i)\}_{i\geq 1}$ *is a sequence of i.i.d. random vectors with* $\mathrm{E}[X_j(i)] < \infty$ *for all* $j \in \{1, \ldots, N\}$ *and* $i \geq 1$. *Then, we have* $T \leq T^{[K,M]}$ *for* $1 \leq K \leq M \leq N$ *if Assumptions 1 and 2 hold, pooling does not affect the distribution of service times at stations that are not pooled, and there are infinite buffers before and after the pooled station.*

Corollary 1 shows that under reasonable conditions on service times and server rates, pooling a subset of neighboring stations in a tandem line will result in an improvement in system throughput when the buffer spaces around the pooled station are unlimited. Similarly to Propositions 1 and 2, Corollary 1 requires the buffers before and after the pooled station to be large, but unlike those propositions, it does not require the buffers between the stations to be pooled to be zero (at the expense of a weaker result about ordering of throughput rather than departure times). Our next result shows that complete pooling is always better than any form of partial pooling under the same mild conditions on service times and server rates.

Proposition 4 *Suppose that* $\{\mathbf{X}(i)\}_{i\geq 1}$ *is a sequence of i.i.d. random vectors with* $\mathrm{E}[X_j(i)] < \infty$ *for all* $j \in \{1, \dots, N\}$ *and* $i \geq 1$. *Then, we have* $T^{[1,N]} \geq T^{[K,M]}$ *for* $1 \leq K \leq M \leq N$ *if Assumptions 1 and 2 hold and pooling does not affect the distribution of service times at stations that are not pooled.*

Finally, we consider a tandem line with $b_j = \infty$ for all $j = 2, \dots, N$ to demonstrate how much improvement in throughput can be gained by parallel pooling.

Proposition 5 *Suppose that* $b_j = \infty$ *for all* $j \in \{2, \dots, N\}$ *and* $\{\mathbf{X}(i)\}_{i\geq 1}$ *is a sequence of i.i.d. random vectors with* $\mathrm{E}[X_j(i)] < \infty$ *for all* $j \in \{1, \dots, N\}$ *and* $i \geq 1$. *Let* $J \in \{1, \dots, N\}$ *be a bottleneck station, i.e.,* $\mathrm{E}[X_J(1)] \geq \mathrm{E}[X_j(1)]$ *for all* $j = 1, \dots, N$. *Under Assumption 1, pooling station* J *with its neighboring stations could lead to an increase in the system throughput by a factor of the number of stations that are pooled if the servers at the pooled station are identical and pooling does not affect the distribution of service times at stations that are not pooled.*

3 Other performance measures

In this section, we study the effects of parallel pooling on the total number of jobs in the system (commonly known as the work-in-process inventory [WIP] in the manufacturing literature), sojourn times, and holding costs. For a fair comparison between the pooled and original systems in terms of these performance measures, in this section we consider the case where the total number of jobs that enter the original and pooled systems are equal at any given time. In order to guarantee this, we replace the assumption of an infinite supply of jobs with the assumption that there is an exogenous arrival stream at the first station, which is also independent of the service times. Recall that we assume that the size of the input buffer of the first station b_1 is infinite, and hence, arrivals to the system are never blocked. We start by noting that our analytical results from Sect. 2 continue to hold for systems with an arrival stream.

One way to model the arrival process to the first station is to consider the tandem line with an infinite supply of jobs but with a dummy station at the front of the line (called station 0), where the service times are equal to the interarrival times between two consecutive jobs and the output buffer has infinite capacity. For all $1 \leq K \leq M \leq N$, let $X_0(i)$ and $X_0^{[K,M]}(i)$ be the times between the $(i-1)$st and ith arrivals at the original and pooled lines, respectively. We then immediately obtain that Proposition 1 still holds under the assumption of an arrival stream at the first station if $X_0^{[K,M]}(i) \leq X_0(i)$ for all $i \geq 1$. Similarly, Proposition 2 can be extended to the case with arrivals under the condition that $\{X_0^{[K,M]}(i)\}_{i\geq 1} \leq_{st} \{X_0(i)\}_{i\geq 1}$ and the assumption that the arrival process is independent of the service time process in both systems. Finally, if the interarrival times are i.i.d. with finite mean and $b_1 = \infty$, Propositions 3, 4, and 5, and Corollary 1 can be shown to hold under the assumption of stochastic arrivals to the first station by a minor modification of their proofs to incorporate the arrival process as a dummy station.

When parallel pooling (stochastically) decreases the departure times from the system with arrivals, then it is easy to show that the total number of jobs in the system (WIP) at any given time (stochastically) decreases, too. However, even when parallel pooling decreases the time between the ith departure from the system and the ith arrival to the system for all $i \geq 1$, it does not always decrease the sojourn time of each job in the system. Since the pooled station has multiple servers, the order the jobs leave from the pooled station (and all the stations downstream) may be different from the order that they enter the pooled station (and all the stations upstream). Hence, as we demonstrate in Example 2 in the Appendix, although the ith departure time from the system is reduced by pooling for all $i \geq 1$, the sojourn time of the ith entering job may actually increase for some $i \geq 1$. Nevertheless, when parallel pooling decreases the total number of jobs in the system (WIP) at any given time, then Little's Law immediately yields that parallel pooling decreases the long-run average sojourn time (if the long-run average sojourn time and number in the system exist; see, for example, page 290 in Kulkarni [16]). Hence, we conclude that whenever parallel pooling decreases the departure times from the system with an arrival stream and hence the total number of jobs at any given time (almost surely or stochastically), then it also decreases the long-run average sojourn time in the system (if it exists).

Finally, we provide a set of conditions under which parallel pooling decreases the total holding costs. Let $h_j \geq 0$ be the holding cost per unit time of a job at station j and at its input buffer for $j = 1, \ldots, N$ in the original line (with arrivals). We assume that when stations $K, \ldots, M$ are parallel pooled, then the holding cost rates $h_1, \ldots, h_{K-1}, h_{M+1}, \ldots, h_N$ at the unpooled stations do not change and we let $h^{[K,M]}$ denote the holding cost rate at the pooled station. Let $H(t)$ and $H^{[K,M]}(t)$ be the total holding costs accumulated during $[0, t]$ for the original and parallel pooled systems, respectively. (Formal definitions of $H(t)$ and $H^{[K,M]}(t)$ are given in the proof of Proposition 6 in the Appendix.)

Proposition 6 *When there is a stochastic arrival stream to station 1, we have* $H^{[K,M]}(t) \leq H(t)$*, for* $t \geq 0$ *and* $1 \leq K \leq M \leq N$*, if*

(i) $D_N^{[K,M]}(i) \leq D_N(i)$ *for all* $i \geq 1$ *such that* $D_N(i) \leq t$*;*
(ii) *if* $K \geq 2$*, then either*
  (a) $h_j = h_K$ *for all* $j = 1, \ldots, K-1$*, or*
  (b) $D_j(i) = D_j^{[K,M]}(i)$ *for all* $j = 1, \ldots, K-1$ *and* $i \geq 1$*;*
(iii) $h_j \geq h_K$ *for* $j = K+1, \ldots, M$*;*
(iv) $h_j = h_K$ *for* $j = M+1, \ldots, N$ *if* $M \leq N-1$*;*
(v) $h^{[K,M]} \leq h_K$.

Proposition 6 shows that pooling several stations in the line will lower the total holding costs if it lowers the departure times from the system (as in Proposition 1), the holding cost rate at each station that is pooled is greater than or equal to the holding cost of the first pooled station, and the holding cost rates at all the other (unpooled) stations are equal to that of the first pooled station. Note that condition $(ii)(b)$ in Proposition 6 holds when $P^{[K,M]} = b_K = \infty$, and pooling does not change service times at stations $1, \ldots, K-1$. Also, Proposition 6 implies that complete pooling always decreases the total holding cost as long as it reduces the departure times from the system and the first station is the cheapest place to store jobs.

4 Teams versus parallel servers

In Sects. 2 and 3, we studied pooling stations when only stations are pooled, not their servers. In an earlier work [4], we studied "cooperative" pooling, where not only stations are pooled but their servers are pooled as well to form a single team that processes jobs at the pooled station. A natural question is then if one has the option of cooperative pooling or parallel pooling, which would be better? In this section, we will provide some analysis to answer this question.

Let $T^{(K,M)}$ represent the steady-state throughput of the line discussed in Sect. 2 where stations K through M are pooled but under cooperative pooling. We first state an assumption on the cooperation of servers when they are pooled.

Assumption 3 The service time of job $i \geq 1$ at the pooled station under cooperative pooling is given by

$$X^{(K,M)}(i) = \sum_{j=K}^{M} \frac{\mu_{j,j} X_j(i)}{\sum_{\ell=K}^{M} \mu_{\ell,j}},$$

for $1 \leq K \leq M \leq N$.

Assumption 3 states that the service rates are *additive*, or equivalently servers neither lose nor gain any efficiency by cooperative pooling. This assumption has been used frequently in the literature on flexible servers (see, for example, [1] and [17]) and is a reasonable assumption when the number of servers to be pooled is small.

Proposition 7 *If Assumptions 1, 2, and 3 hold, and $\{\sum_{j=1}^{N} \theta_j X_j(i)\}_{i \geq 1}$ is a sequence of i.i.d. random variables with finite mean, then we have $T^{(1,N)} = T^{[1,N]}$.*

Proposition 7 implies that pooling all stations in a line with i.i.d. service times at all stations yields the same system throughput under the parallel and cooperative pooling structures given by Assumptions 1 and 3, respectively, if the service rates satisfy the product form of Assumption 2. (A result similar to Proposition 7 is also proved by Van Oyen, Gel, and Hopp [25], but under the assumption of identical servers.) Proposition 7 leads to Corollary 2, which extends the result to the partial pooling case when the input and output buffers of the pooled stations are infinite.

Corollary 2 *Suppose that $\{(\mathbf{X}_{1,K-1}(i), \sum_{j=1}^{N} \theta_j X_j(i), \mathbf{X}_{M+1,N}(i))\}_{i \geq 1}$ is a sequence of i.i.d. random vectors with $\mathrm{E}[X_j(i)] < \infty$ for all $j = 1, \ldots, N$ and $i \geq 1$. Then, we have $T^{(K,M)} = T^{[K,M]}$ if Assumptions 1, 2, and 3 hold, pooling does not affect the distribution of service times at stations that are not pooled, and there are infinite buffers before and after the pooled station under both the parallel and cooperative pooling structures.*

The main insight that we obtain from Proposition 7 and Corollary 2 is that if the buffer sizes around the pooled station are not limited, then it does not matter whether one chooses parallel or cooperative pooling. The intuition is that if pooling does not impact the departure times at the stations that are upstream from the pooled station (because the upstream service times are unaffected and $P^{[K,M]}$ is infinite) and if the servers at the pooled station never have to idle due to blocking (because $Q^{[K,M]}$ is infinite), then the departure rate from the pooled station would be the same (under Assumptions 1, 2, and 3) whether it is obtained by parallel pooling or cooperative pooling. However, when the pooled station can be blocked or can block other stations, then it does matter whether it is obtained by cooperative or parallel pooling, as we see in the remainder of this section. Note that in parallel pooled stations, a blocked server cannot help another server at the same station but under cooperative pooling all pooled servers will work together as a team until the entire station is blocked.

Consider now a tandem line of two stations with an infinite supply of jobs and a finite buffer between the two stations. Suppose that jobs at each station have i.i.d. service times that come from an exponential distribution with mean one and that there are $L_i \geq 2$ identical servers at station i with μ_i being the rate of a single server at station i, for $i = 1, 2$. We will consider this system under four configurations. In System 0, none of the servers are pooled, which means that all L_i servers are working in parallel at station $i \in \{1, 2\}$. In System $i \in \{1, 2\}$, servers at station i work cooperatively with additive service rates (i.e., there is a single server at station i with rate $L_i\mu_i$), whereas servers at station $3 - i$ work in parallel. Finally, in System 3, servers at each station work cooperatively with additive service rates, i.e., the system is a tandem line with a single server at station $i \in \{1, 2\}$ working at a rate of $L_i\mu_i$. Note that the level of cooperation increases from System 0 to Systems 1 and 2, and then further from Systems 1 and 2 to System 3.

It is well-known that buffer capacities affect throughput. In particular, increasing the buffer sizes would increase the system throughput in most tandem networks (see, for example, Glasserman and Yao [11]). In that respect, when we compare two lines with cooperative servers and parallel servers, with everything else in the networks being the same, the system with parallel servers has an advantage. This is because each individual server also acts as a storage space, and hence if the buffers between two stations have the same size, then the system with parallel servers will have a larger number of storage spaces than the system with cooperative servers. Therefore, when we compare Systems 0, 1, 2, and 3, we allow them to have different buffer sizes between the two stations, and thus allow the buffer size to be another design parameter in their comparison. For System $j \in \{0, 1, 2, 3\}$, let B_j, where $0 \leq B_j < \infty$, be the number of buffers between stations 1 and 2 and let T_j be the steady-state throughput.

It is easy to see that the four systems under consideration can be modeled as birth–death processes with different birth and death rates. We can then compare them in terms of their steady-state system throughput as stated in the following proposition.

Proposition 8 *For fixed $j \in \{1, 2\}$, we have*

(i) $T_0 < T_j$ *if* $B_j \geq B_0 + L_j - 1$ *and* $T_j < T_0$ *if* $B_j \leq B_0$*;*
(ii) $T_j < T_3$ *if* $B_3 \geq B_j + L_{3-j} - 1$ *and* $T_3 < T_j$ *if* $B_3 \leq B_j$*.*

Proposition 8 implies that if the number of buffers in the pooled system is sufficiently large, then higher levels of cooperation yield strictly better throughput. For example, suppose that the B_j are chosen for $j \in \{1, 2, 3\}$ such that all four system configurations have the same total physical space as in System 0, i.e., $L_1 + L_2 + B_0$ physical spaces, by letting $B_j = B_0 + L_j - 1$ for $j \in \{1, 2\}$, and $B_3 = B_0 + L_1 + L_2 - 2$. In this case, by Proposition 8, System 0 provides the smallest and System 3 provides the largest throughput, whereas Systems 1 and 2 yield performance in between Systems 0 and 3. This shows that having cooperative servers yields a larger throughput than having parallel servers when the two systems are equal in terms of the total amount of physical space. Note that at a station with cooperative servers, all servers can work until no job can be processed at that station due to blocking or idling. However, for a similar situation at a system with parallel servers, it is possible that some servers at a station work while other servers at the same station stay idle. This improvement by cooperative servers in tandem lines with finite buffers is in contrast with the results on tandem lines with infinite buffers (such as Corollary 2), where having parallel or cooperative servers (with the same total service capacity) does not affect the steady-state throughput. On the other hand, Proposition 8 also implies that if the B_j for $j \in \{1, 2, 3\}$ are all set to B_0 (i.e., the number of buffers in System 0), then System 3 provides the smallest and System 0 provides the largest throughput, whereas Systems 1 and 2 again provide performance in between Systems 0 and 3. This means that the advantage of cooperative servers may no longer hold if the systems are not equal in terms of total physical spaces. More specifically, if additional buffers cannot be added to the system with cooperative pooling, then the system with parallel servers will be more beneficial because of the extra storage space that each server provides.

5 Numerical results

With the objective of quantifying the possible improvements obtained by parallel pooling and gaining better insights about when and how this approach should be used, we have conducted a number of numerical experiments. In particular, we have studied the effects of parallel pooling on the steady-state throughput and WIP of tandem lines with three and four stations.

Recall that in Sect. 2, we obtained a set of conditions under which parallel pooling improves the departure process; see Propositions 1 and 2. One of these conditions was that the service time of each job at each station in the pooled system should not be larger than the corresponding service time in the original system, and another condition was that there should be zero buffers between the stations that are pooled. In this section, one of our main goals is to provide evidence suggesting that parallel pooling can still improve system throughput when there are buffers between the pooled stations (as long as these buffers are allocated properly) and when pooling causes longer service times at the pooled stations, for example, because servers may need additional time to switch between different tasks. Numerical results in this section will also provide insights into the magnitude of gain obtained by parallel pooling and its comparison with that under cooperative pooling.

Throughout this section, we assume that all servers are identical, service times at station $j \in \{1, \ldots, N\}$ are exponentially distributed with rate $\gamma_j \geq 0$, and service times are independent across jobs and stations. We also assume that there is an infinite supply of jobs in front of the first station (we focus on this case, rather than outside arrivals, because the main performance measure of interest in this paper is the steady-

Table 1 Throughput (THP) and WIP of balanced lines with $N \in \{3, 4\}$ and $b_j = 0$, for $j = 2, \ldots, N$, after parallel pooling

System	THP	% Inc. in THP	WIP	% Inc. in WIP	% Dec. in WIP	$\overline{\beta}$
$N = 3$						
1-2-3	0.5641	–	2.3590	–	–	–
(12)-3	0.7290	29.23	2.7290	15.68	–	1.51
1-(23)	0.7290	29.23	2.4580	4.20	–	1.51
(123)	1.0000	77.27	3.0000	27.17	–	1.77
$N = 4$						
1-2-3-4	0.5148	–	3.0646	–	–	–
(12)-3-4	0.5990	16.36	3.4562	12.78	–	1.48
1-(23)-4	0.6268	21.76	3.2700	6.70	–	1.47
1-2-(34)	0.6080	18.10	2.9690	–	3.12	1.53
(123)-4	0.7570	47.05	3.7570	22.59	–	1.77
1-(234)	0.7570	47.05	3.2709	6.73	–	1.77
(1234)	1.0000	94.25	4.0000	30.52	–	1.94

state throughput). When stations K through M are pooled, we assume that the service time of a job at the pooled station is equal to the sum of $M-K+1$ exponential random variables with means $\beta\gamma_j^{-1}$, for $j = K, \ldots, M$ and some scaling factor $\beta \geq 1$. With the introduction of the scaling parameter β in our numerical study, we can observe how much of an increase in service times by pooling is tolerable for pooling to be still beneficial in terms of enhancing the throughput. Note that $\beta > 1$ corresponds to the case where the service times at the pooled station increase by pooling, whereas $\beta = 1$ represents the case where they do not change.

We first consider balanced lines (where the service requirements are i.i.d. at all stations before pooling, and hence there is no bottleneck station that is slower than the other stations) with $\gamma_j = 1.0$ for $j \in \{1, \ldots, N\}$ and $N \in \{3, 4\}$; see Tables 1 and 2. To specify different system configurations, we use hyphens to separate the stations, put the pooled stations between parentheses, and denote each buffer space with a small letter "b". For example, when $N = 4$ and $b_2 = b_3 = b_4 = 3$, then 1-bbb2-bbb3-bbb4 denotes the original system and 1-$bbbb$(23)-$bbbbb$4 denotes the system for which stations 2 and 3 are pooled, $P^{[2,3]} = 4$, and $Q^{[2,3]} = 5$. In Tables 1 and 2, the second and fourth columns, respectively, provide the steady-state throughput and WIP for different parallel pooling structures with $\beta = 1$ for lines with $N \in \{3, 4\}$ and common buffer sizes $b_j \in \{0, 3\}$ for $j \in \{2, \ldots, N\}$. We also provide the percentage increase in throughput and percentage decrease/increase in WIP obtained over the original line by each pooling structure with $\beta = 1$ in Tables 1 and 2. Finally, in the last column, we present the largest value of β under which the specified pooling structure would increase the long-run average throughput (denoted by $\overline{\beta}$). For complete pooling, it is not difficult to see that the throughput under scaling parameter β equals $T^{[1,N]}/\beta$, and hence $\overline{\beta} = T^{[1,N]}/T$. For partial pooling, we identify the value of $\overline{\beta}$ numerically.

We can summarize our conclusions on parallel pooling from Tables 1 and 2 as follows:

Table 2 Throughput (THP) and WIP of balanced lines with $N \in \{3, 4\}$ and $b_j = 3$, for $j = 2, \ldots, N$, after parallel pooling

System	THP	% Inc. in THP	WIP	% Inc. in WIP	% Dec. in WIP	$\overline{\beta}$
$N = 3$						
1-*bbb*2-*bbb*3	0.7767	–	5.7001	–	–	–
(12)-*bbbbbb*3	0.9140	17.67	6.0311	5.81	–	1.26
1-*bbbbbb*(23)	0.9140	17.67	5.7109	0.19	–	1.28
(123)	1.0000	28.75	3.0000	–	47.37	1.28
$N = 4$						
1-*bbb*2-*bbb*3-*bbb*4	0.7477	–	8.0813	–	–	–
(12)-*bbbbbb*3-*bbb*4	0.8225	10.00	9.5895	18.66	–	1.28
1-*bbb*(23)-*bbbbbb*4	0.8438	12.85	7.4079	–	8.33	1.25
1-*bbbb*(23)-*bbbbb*4	0.8511	13.83	7.9773	–	1.29	1.27
1-*bbbbb*(23)-*bbbb*4	0.8511	13.83	8.4934	5.10	–	1.27
1-*bbbbbb*(23)-*bbb*4	0.8438	12.85	9.0346	11.80	–	1.25
1-*bbb*2-*bbbbbb*(34)	0.8232	10.09	6.6745	–	17.41	1.28
(123)-*bbbbbbbbb*4	0.9428	26.09	8.6662	7.24	–	1.33
1-*bbbbbbbbb*(234)	0.9428	26.09	8.1048	0.29	–	1.33
(1234)	1.0000	33.74	4.0000	–	50.50	1.33

1. When pooling does not increase mean service times (i.e., $\beta = 1$), parallel pooling any group of adjacent stations in a balanced line improves the system throughput regardless of the buffer allocation around the pooled station. Moreover, this improvement in throughput in balanced lines is substantial, falling in the range of 10.00–94.25% when $N \in \{3, 4\}$.
2. In all cases considered, pooling is beneficial even when it leads to 25% longer service times. This tolerance for longer service times is even larger for systems with smaller buffers and with a larger number of pooled stations.
3. The more stations are pooled, the better the throughput gets. Also, systems with the same number of stations after pooling provide similar throughput.
4. Pooling stations near the middle of the line yields better throughput than pooling those at the beginning or end of the line when systems with the same number of pooled stations are compared.
5. Parallel pooling several stations at the end of the line provides slightly better throughput than parallel pooling several stations at the beginning of the line if there are more than two stations in the pooled system (for example, compare pooled systems (12)-3-4 and 1-2-(34) in Table 1). This is consistent with Hillier and So [13], who provide numerical results that support the fact that placing any extra servers at the last station in a tandem line provides slightly better throughput than placing these extra servers at the first station.
6. Partial parallel pooling (i.e., parallel pooling only a subset of the stations in the tandem line) generally increases the WIP in balanced lines. (This does not contradict our conclusion in Sect. 3 because of the differences in the assumption about

job arrivals in the two sections.) One exception is when several stations at the end of the line are pooled or more buffers are allocated toward the end of the line, in which case jobs may be pushed out of the system more efficiently.

Tables 1 and 2 also provide useful insights on the comparison of parallel and cooperative pooling structures when compared with Tables 1 and 2 in Argon and Andradóttir [4]. One important observation is that all of the above listed conclusions for parallel pooling under $\beta = 1$ also hold for cooperative pooling, except for items 5 and 6. For cooperative pooling, pooling at the beginning or end of the line yields the same throughput in a balanced line due to the reversibility principle of tandem lines with a single server at each station. Note that the reversibility principle for tandem lines with multiple parallel servers holds if and only if there are two stations in the system; see, for example, Theorem 4 in Yamazaki et al. [26]. Also, cooperative and parallel pooling structures differ with respect to their effects on WIP. In particular, parallel pooling increases WIP in more scenarios than cooperative pooling does when lines with the same pooled stations and the same number of total physical spaces are compared. Moreover, parallel pooling seems to provide a smaller throughput than cooperative pooling in most cases, which is consistent with Proposition 8. For example, in the balanced line with four stations and zero buffers, parallel pooling the first three stations provides approximately 10% smaller throughput than the corresponding cooperative pooling structure (0.7570 vs. 0.8421). Note, however, that the difference between the throughputs of the pooled system with cooperative servers and the pooled system with parallel servers diminishes for larger buffer sizes. This is consistent with Corollary 2, which proves which parallel pooling and cooperative pooling provide the same throughput when there are infinite buffers around the pooled station. The only cases where parallel pooling provides the same or slightly better throughput are when all stations in the line are pooled or when all buffers between the stations that are pooled in the original line are added only to one side of the pooled station (for example, for 1-*bbb*(23)*bbbbbb*-4), respectively. Finally, parallel pooling seems to provide consistently higher WIP than cooperative pooling. This makes intuitive sense because a larger number of jobs in service is needed by a station with parallel servers to achieve a similar service capacity with a station having cooperative servers.

We next look at the effects of parallel pooling on the steady-state throughput and WIP of unbalanced tandem lines with four stations. For these tandem lines, we generate the service rate γ_j at each station $j \in \{1, 2, 3, 4\}$ independently from a uniform distribution on the range $[0.1, 20.1]$. We consider both lines that have the same amount of buffer spaces between any two stations (i.e., $b_2 = b_3 = b_4 \in \{0, 3\}$) and lines for which the buffers between any two stations are generated independently from a discrete uniform distribution on the set $\{0, 1, 2, 3\}$. Using this experimental setting, we generate 5000 lines independently and provide a summary of the results for $\beta \in \{1, 1.25\}$ in Tables 3 and 4, respectively. In particular, based on these 5000 instances, we estimate the probability of observing an increase in the system throughput and WIP, and for those cases in which parallel pooling increases the system throughput, we estimate a 95% confidence interval on the percentage increase in throughput over the unpooled system. Confidence intervals on percentage decrease in throughput and percentage increase/decrease in WIP are computed similarly.

Table 3 Throughput (THP) and WIP of unbalanced lines with $N = 4$, after parallel pooling with $\beta = 1$

System	Prob. of Incr. in THP	% Incr. in THP	% Decr. in THP	Prob. of Incr. in WIP	% Incr. in WIP	% Decr. in WIP
Common buffer size = 0						
(12)-3-4	1.0000	27.20 ± 0.75	–	1.0000	17.73 ± 0.60	–
1-(23)-4	1.0000	29.68 ± 0.72	–	0.8406	10.58 ± 0.38	2.19 ± 0.17
1-2-(34)	1.0000	28.40 ± 0.76	–	0.3100	11.68 ± 0.50	3.94 ± 0.12
(123)-4	1.0000	75.37 ± 1.35	–	1.0000	34.87 ± 1.04	–
1-(234)	1.0000	75.20 ± 1.37	–	0.6828	24.02 ± 0.77	4.75 ± 0.23
(1234)	1.0000	148.59 ± 1.35	–	1.0000	52.53 ± 1.44	–
Common buffer size = 3						
(12)-B3-4	1.0000	25.37 ± 0.83	–	0.8004	25.44 ± 1.12	34.39 ± 1.08
1-(23)-B4	0.9936	25.34 ± 0.81	0.0006 ± 0.0002	0.4310	9.41 ± 0.45	17.87 ± 0.46
1-B(23)-4	0.9960	25.28 ± 0.81	0.0005 ± 0.0002	0.6876	21.97 ± 0.89	10.55 ± 0.49
1-(23)-4*	1.0000	25.16 ± 0.81	–	0.6140	8.49 ± 0.34	10.75 ± 0.43
1-2-B(34)	1.0000	25.22 ± 0.84	–	0.2556	23.82 ± 0.83	11.70 ± 0.39
(123)-B4	1.0000	64.77 ± 1.50	–	0.5556	48.82 ± 2.39	39.14 ± 0.76
1-B(234)	1.0000	64.51 ± 1.52	–	0.5522	54.13 ± 1.75	21.34 ± 0.73
(1234)	1.0000	117.74 ± 1.83	–	0.1790	106.58 ± 5.40	48.63 ± 0.52
Buffer sizes $\sim$ uniform{0, 1, 2, 3}						
(12)-B3-4	1.0000	27.48 ± 0.82	–	0.8708	23.48 ± 0.89	25.67 ± 1.29
1-(23)-B4	0.9978	28.09 ± 0.79	0.10 ± 0.09	0.5222	11.50 ± 0.50	14.05 ± 0.49
1-B(23)-4	0.9976	28.12 ± 0.79	0.06 ± 0.05	0.7320	20.71 ± 0.98	7.49 ± 0.42
1-(23)-4*	1.0000	28.00 ± 0.78	–	0.6804	10.39 ± 0.40	7.78 ± 0.38
1-2-B(34)	1.0000	27.65 ± 0.83	–	0.2534	23.56 ± 1.23	9.72 ± 0.31
(123)-B4	1.0000	71.68 ± 1.45	–	0.6866	43.02 ± 1.71	28.19 ± 0.84
1-B(234)	1.0000	71.41 ± 1.47	–	0.5678	50.78 ± 2.04	14.78 ± 0.58
(1234)	1.0000	128.57 ± 1.67	–	0.3518	73.31 ± 3.37	33.92 ± 0.59

In Tables 3 and 4, we use a capital letter "B" to indicate the location where the buffers between the pooled stations are placed. When the buffer sizes are positive, then there are more than two buffer allocation schemes to consider when stations 2 and 3 are pooled (for example, if there are two buffers between stations 2 and 3, then we can either place the two buffers before or after the pooled station or place one buffer before and the other buffer after the pooled station). Among all possible alternatives, we only consider placing all buffers before or after the pooled station. Moreover, we also consider the pooled system in which all buffers between stations 2 and 3 are placed before (after) the pooled station if station 3 (2) is slower than station 2 (3); we denote this system by 1-(23)-4*. We consider this particular buffer allocation structure since it corresponds to the buffer allocation scheme that we have recommended for cooperative pooling based on Proposition 1 of Argon and Andradóttir [4] (i.e., placing the pooled station at the position of the slowest station among the stations that are pooled). Note that there is a rich literature on the optimal buffer allocation problem in

Table 4 Throughput (THP) and WIP of unbalanced lines with $N = 4$, after parallel pooling with $\beta = 1.25$

System	Prob. of Incr. in THP	% Incr. in THP	% Decr. in THP	Prob. of Incr. in WIP	% Incr. in WIP	% Decr. in WIP
Common buffer size = 0						
(12)-3-4	1.0000	14.95 ± 0.43	–	1.0000	14.34 ± 0.56	–
1-(23)-4	0.9760	16.53 ± 0.41	0.02 ± 0.05	0.9734	9.93 ± 0.33	1.57 ± 0.22
1-2-(34)	1.0000	16.06 ± 0.43	–	0.8458	7.59 ± 0.25	1.94 ± 0.14
(123)-4	1.0000	49.81 ± 0.91	–	1.0000	32.58 ± 1.02	–
1-(234)	1.0000	49.78 ± 0.92	–	0.9032	22.42 ± 0.63	3.18 ± 0.32
(1234)	1.0000	98.87 ± 1.08	–	1.0000	52.53 ± 1.44	–
Common buffer size = 3						
(12)-B3-4	0.9446	12.87 ± 0.49	2.13 ± 0.14	0.6666	17.12 ± 0.94	31.40 ± 0.88
1-(23)-B4	0.8716	13.66 ± 0.51	0.98 ± 0.09	0.4808	6.21 ± 0.31	17.03 ± 0.47
1-B(23)-4	0.8860	13.45 ± 0.50	1.05 ± 0.10	0.8552	18.77 ± 0.75	5.55 ± 0.54
1-(23)-4*	0.9416	12.60 ± 0.48	1.70 ± 0.13	0.7348	5.83 ± 0.22	6.16 ± 0.33
1-2-B(34)	0.9524	12.79 ± 0.49	2.17 ± 0.15	0.5974	15.25 ± 0.57	5.61 ± 0.35
(123)-B4	1.0000	39.09 ± 1.04	–	0.4602	42.26 ± 2.34	40.52 ± 0.66
1-B(234)	1.0000	39.10 ± 1.05	–	0.7300	48.82 ± 1.50	13.83 ± 0.74
(1234)	1.0000	74.19 ± 1.47	–	0.1790	106.58 ± 5.40	48.63 ± 0.52
Buffer sizes $\sim$ uniform{0, 1, 2, 3}						
(12)-B3-4	0.9910	14.65 ± 0.46	2.02 ± 0.36	0.7818	16.84 ± 0.76	23.03 ± 0.99
1-(23)-B4	0.9642	15.10 ± 0.46	1.03 ± 0.17	0.6062	8.76 ± 0.39	14.64 ± 0.56
1-B(23)-4	0.9640	15.15 ± 0.46	1.08 ± 0.17	0.8978	17.79 ± 0.83	4.70 ± 0.50
1-(23)-4*	0.9914	14.63 ± 0.44	1.56 ± 0.42	0.8244	7.98 ± 0.30	5.07 ± 0.35
1-2-B(34)	0.9946	14.83 ± 0.47	1.90 ± 0.39	0.6208	13.97 ± 0.69	4.99 ± 0.27
(123)-B4	1.0000	45.33 ± 0.98	–	0.6048	38.47 ± 1.71	28.41 ± 0.75
1-B(234)	1.0000	45.31 ± 0.99	–	0.7746	44.16 ± 1.66	10.30 ± 0.63
(1234)	1.0000	82.86 ± 1.34	–	0.3518	73.31 ± 3.37	33.92 ± 0.59

finite-capacity tandem networks; see, for example, [11,14,23]. Since the main focus of this paper is observing the effects of pooling, we do not seek the best buffer allocation design but instead identify simple buffer allocation structures under which pooling improves the steady-state throughput.

Tables 3 and 4 show that pooling generally improves throughput in unbalanced lines, even when it results in larger service times, and that the benefit is larger when more stations are pooled. From Table 3, one can observe that parallel pooling several stations at the beginning or end of a line always improves the system throughput when $\beta = 1$ regardless of the buffer sizes in the system. Moreover, parallel pooling any group of stations improves the system throughput if there are zero buffers in the system. On the other hand, when the buffers between at least some of the stations are positive, then pooling intermediate stations may decrease the system throughput if the buffers are not allocated properly around the pooled station. (Note that this result is in agreement with Example 1, which suggests that the conditions on the buffers in Proposition 1 are at least to some extent necessary.) If the buffer allocation is performed as in system 1-(23)-4*, then intermediate pooling improves the throughput in all 5000 instances. However, even if the buffer allocation is not done properly, intermediate pooling decreases the throughput only very rarely and the amount of decrease is marginal. Indeed, when we first designed the experiment presented in Table 3, we used the same range for the uniform distribution of service rates, namely, [0.5, 2.5], used in the corresponding experiment for cooperative pooling by Argon and Andradóttir [4] presented in their Table 4. However, for that experiment, none of the 5000 instances resulted in a decrease in throughput by parallel pooling the middle stations. Hence, we had to use a wider range of service rates (i.e., $[0.1, 20.1]$) to create highly unbalanced lines in order to observe lines where parallel pooling would decrease throughput due to poor buffer allocation. This suggests that buffer allocation is less of a concern for parallel pooling compared to cooperative pooling.

Table 3 also provides insights into the comparison of parallel and cooperative pooling structures when compared to Table 6 of Argon and Andradóttir [4], which uses the same range for the uniformly distributed service rates, namely, $[0.1, 20.1]$. In particular, these two tables present results on the same set of numerical experiments except that one applies parallel pooling, whereas the other employs cooperative pooling without adding extra buffers to equate the total number of storage spaces. This comparison shows that parallel pooling results in a larger fraction of instances where pooling increases throughput than cooperative pooling (without added storage spaces) but at a cost of degradation in WIP. However, when either form of pooling increases the throughput, the average percentage increase is similar.

Finally, from Table 4, we observe that even when pooling causes a 25% increase in mean service times at the pooled stations, only a small fraction of the unbalanced lines generated had a degradation in throughput by pooling. This rare reduction in throughput happened mostly by pooling intermediate stations and it was no larger than 2.2% on average. WIP appears to be more likely to increase by pooling under $\beta = 1.25$ when compared to the case with $\beta = 1$ except when stations at the beginning are pooled. On the other hand, the percentage change in WIP is always smaller when the WIP increases and usually smaller when the WIP decreases (except when three stations are pooled at the beginning of the line) as compared to the case where pooling does not change the mean service times.

6 Conclusions

For a tandem network of single-server queues with finite buffers, general service times, and flexible, but non-collaborative, servers, we have considered *parallel pooling* several stations with the objective of improving the system throughput. We first provided sufficient conditions on the service times and buffers under which parallel pooling several stations permanently decreases the departure times from the system and hence increases the steady-state system throughput. More specifically, we have shown analytically that if the service time of each job at the pooled station is no larger than the sum of the service times at the stations that are pooled and there are no buffers between the stations that are pooled, then parallel pooling will result in earlier departures from the system. Our numerical results on lines with three and four stations suggest that parallel pooling in a system with identical servers generally improves the system throughput even when there are buffers between the pooled stations in the original line and pooling results in longer service times at the pooled stations. Furthermore, this improvement by parallel pooling can be substantial and is increasing in the number of stations pooled.

In this article, we also compared the effects of having multiple parallel servers versus a pooled team of cooperative servers on the throughput of tandem lines. Our analytical and numerical results suggest that when the maximal WIP capacity of a line (including the spaces allocated for service and waiting) is finite and constant, then in most cases having cooperative servers results in a larger throughput than having parallel servers under the assumption that servers are identical and service rates are additive. However, if pooling servers into teams results in a reduction of physical spaces where jobs could be stored, then having parallel servers is more likely to yield a higher throughput.

Acknowledgements The work of the first author was supported by the National Science Foundation under Grants DMI-0000135, CMMI-1234212, and CMMI-1635574. The work of the second author was supported by the National Science Foundation under grants DMI-0000135 and CMMI-1536990. We thank two anonymous referees for comments that led to substantial improvements in the paper.

Appendix

In this appendix, we provide proofs of our theoretical results, lemmas that are used in some of our proofs, and other supplementary material. We use Lemmas 1 and 2 to prove Proposition 1. The proof of Lemma 1 is trivial and hence is omitted.

Lemma 1 *If a_i and b_i are some real numbers for $i = 1, \ldots, n$, where n is a positive integer, then we have*

$$\max_{i=1,\ldots,n} \{a_i\} - \max_{i=1,\ldots,n} \{b_i\} \geq \min_{i=1,\ldots,n} \{a_i - b_i\} .$$

Lemma 2 *Let $\{a_i\}_{i=1}^n$ be a sequence of real numbers, where n is a positive integer. Then, for $J \in \{2, \ldots, n\}$, $k \in \{1, \ldots, J-1\}$, and $\ell_j \in \{1, \ldots, n\}$, for all $j \in \{1, \ldots, k\}$, we have*

$$\Phi_J^{(n)} \{a_i : i \in \{1, \ldots, n\}\} \leq \Phi_{J-k}^{(n-k)} \{a_i : i \in \{1, \ldots, n\} \backslash \{\ell_1, \ldots, \ell_k\}\} .$$

Proof of Lemma 2 Let $m \leq k$ be the number of elements in $\{a_{\ell_1}, \ldots, a_{\ell_k}\}$ that are greater than $\Phi_J^{(n)} \{a_i : i \in \{1, \ldots, n\}\}$. Then,

$$\begin{aligned} \Phi_J^{(n)} \{a_i : i \in \{1, \ldots, n\}\} &= \Phi_{J-m}^{(n-k)} \{a_i : i \in \{1, \ldots, n\} \backslash \{\ell_1, \ldots, \ell_k\}\} \\ &\leq \Phi_{J-k}^{(n-k)} \{a_i : i \in \{1, \ldots, n\} \backslash \{\ell_1, \ldots, \ell_k\}\} , \end{aligned}$$

where the equality holds because when m elements that are larger than $\Phi_J^{(n)}\{a_i : i \in \{1,\ldots,n\}\}$ are taken out from the set, then the Jth largest element becomes the $(J-m)$th largest in the new set. □

Proof of Proposition 1 For $j \in \{1,\ldots,K-1,M,\ldots,N\}$ and $i \geq 1$, let $\Delta_j(i) = D_j(i) - D_j^{[K,M]}(i)$. Also, let $\Delta_j(i) = D_M(i) - A_{j-K+1}(i-M+j+1)$ for $j \in \{K,\ldots,M-1\}$ and $i \geq 1$. For convenience, assume that $\Delta_j(i) = 0$ when $j \notin \{1,\ldots,N\}$ or $i \leq 0$. Consider now the following inequalities for $i \geq 1$:

$$\Delta_j(i) \geq \min\left\{\Delta_{j-1}(i), \Delta_j(i-1), \Delta_{j+1}\left(i-b_{j+1}-1\right)\right\}, \quad \forall j \in \{1,\ldots,K-2,M+1,\ldots,N\}; \tag{8}$$

$$\Delta_{K-1}(i) \geq \min\left\{\Delta_{K-2}(i), \Delta_{K-1}(i-1), \Delta_M\left(i-P^{[K,M]}-M+K-1\right)\right\}; \tag{9}$$

$$\Delta_K(i) \geq \min\left\{\Delta_{K-1}(i), \Delta_M(i-M+K-1), \Delta_K(i-1)\right\}; \tag{10}$$

$$\Delta_j(i) \geq \Delta_{j-1}(i), \ \forall j \in \{K+1,\ldots,M-1\}; \tag{11}$$

$$\Delta_M(i) \geq \min\left\{\Delta_{M-1}(i), \Delta_{M+1}\left(i-Q^{[K,M]}-1\right)\right\}. \tag{12}$$

It is easy to see that the inequalities (8) through (12) imply that $\Delta_j(i) \geq 0$ for all $i \geq 1$ and $j \in \{1,\ldots,N\}$. Then, it remains to show that the inequalities (8) through (12) are true.

We first provide a recursive formula that the departure times $D_j(i)$ must satisfy. For convenience, we assume that $D_j(i) = X_j(i) = 0$ if $j \notin \{1,\ldots,N\}$ or $i \leq 0$. Then, for all $i \geq 1$, we have

$$D_j(i) = \max\left\{D_{j-1}(i)+X_j(i), D_j(i-1)+X_j(i), D_{j+1}\left(i-b_{j+1}-1\right)\right\}, \forall j \in \{1,\ldots,N\}. \tag{13}$$

Now, using condition (i), Lemma 1, and Eqs. (2) and (13) gives inequality (8). Similarly, using condition (i), Lemma 1, and Eqs. (3) and (13), we obtain

$$\Delta_{K-1}(i) \geq \min\left\{\Delta_{K-2}(i), \Delta_{K-1}(i-1), D_K(i-b_K-1) - D_M^{[K,M]}\left(i-P^{[K,M]}-M+K-1\right)\right\}.$$

Then, using Eq. (13) and the condition that $b_j = 0$ for $j \in \{K+1,\ldots,M\}$ iteratively yields $D_K(i-b_K-1) \geq D_M(i-b_K-M+K-1)$ for all $i \geq 1$. The condition that $P^{[K,M]} \geq b_K$ now yields $D_K(i-b_K-1) \geq D_M(i-P^{[K,M]}-M+K-1)$ for all $i \geq 1$, which completes the proof of inequality (9).

Next, we prove inequality (10). Since $A_1(i) \geq A_j(i)$ for all $i \geq 1$ and $j \in \{1,\ldots,M-K\}$, Eq. (6) gives

$$A_1(i+1) = \max\Big\{D_{K-1}^{[K,M]}(i+M-K) + X^{[K,M]}(i+M-K),$$
$$D_M^{[K,M]}(i-1) + X^{[K,M]}(i+M-K), A_1(i)\Big\},$$

for all $i \geq 1$. Then, it is easy to obtain that

$$\Delta_K(i) = \min\Big\{D_M(i) - D_{K-1}^{[K,M]}(i) - X^{[K,M]}(i), D_M(i) - D_M^{[K,M]}(i-M+K-1)$$
$$-X^{[K,M]}(i), D_M(i) - A_1(i-M+K)\Big\}. \tag{14}$$

It now follows from condition (ii) and the fact that $D_M(i) \geq D_{K-1}(i) + \sum_{j=K}^{M} X_j(i)$ for all $i \geq 1$ that the first term of the minimum operator in Eq. (14) is greater than or equal to $\Delta_{K-1}(i)$. Similarly, note that $D_M(i) \geq D_K(i) + \sum_{j=K+1}^{M} X_j(i) \geq D_K(i-1) + \sum_{j=K}^{M} X_j(i)$, for all $i \geq 1$, so that condition (ii) implies that the second term of the minimum operator in Eq. (14) is greater than or equal to $D_K(i-1) - D_M^{[K,M]}(i-M+K-1)$, for all $i \geq 1$. Moreover, using Eq. (13) and the condition that $b_j = 0$ for $j \in \{K+1, \ldots, M\}$ iteratively, one can obtain that $D_K(i-1) \geq D_M(i-M+K-1)$ and hence that the second term of the minimum operator in Eq. (14) is greater than or equal to $\Delta_M(i-M+K-1)$. Noting that $D_M(i) \geq D_M(i-1)$ for all $i \geq 1$ yields inequality (10).

We next prove inequality (11). Using Lemma 2 with $k = j-1$ and Eq. (6), we have

$$A_j(i+1) \leq \max\left\{A_{j-1}(i), \ldots, A_{M-K}(i)\right\} = A_{j-1}(i),$$

for $j \in \{2, \ldots, M-K\}$ and $i \geq 1$. Then, inequality (11) is immediate. Finally, we show that inequality (12) is true. Equation (5) implies that $D_M^{[K,M]}(i) \leq \max\{A_{M-K}(i), D_{M+1}^{[K,M]}(i - Q^{[K,M]} - 1)\}$, and hence that

$$\Delta_M(i) \geq \min\left\{D_M(i) - A_{M-K}(i), D_M(i) - D_{M+1}^{[K,M]}\left(i - Q^{[K,M]} - 1\right)\right\}, \tag{15}$$

for all $i \geq 1$. Note that the first term of the minimum operator in inequality (15) is equal to $\Delta_{M-1}(i)$, for all $i \geq 1$. Moreover, using Eq. (13) and the condition that $Q^{[K,M]} \geq b_{M+1}$, we obtain that $D_M(i) \geq D_{M+1}(i - Q^{[K,M]} - 1)$, for all $i \geq 1$, which immediately yields that the second term of the minimum operator in inequality (15) is greater than or equal to $\Delta_{M+1}(i - Q^{[K,M]} - 1)$ for all $i \geq 1$, and the proof is complete. □

To prove Proposition 2, we need the following lemma, whose proof is immediate.

Lemma 3 *Let $\mathcal{Y} = \{\mathbf{Y}(i)\}_{i\geq 1}$ and $\mathcal{Z} = \{\mathbf{Z}(i)\}_{i\geq 1}$ be two stochastic processes. If $\mathcal{Y} \leq_{st} \mathcal{Z}$, then $\phi(\mathcal{Y}) \leq_{st} \phi(\mathcal{Z})$ for every non-decreasing functional $\phi : \mathbb{R}^\infty \to \mathbb{R}^\infty$.*

Proof of Proposition 2 Let $\phi : \mathbb{R}_+^\infty \to \mathbb{R}_+^\infty$ be defined by $\{\mathbf{D}^{[K,M]}(i)\}_{i\geq 1} = \phi(\{\mathbf{X}^{[K,M]}(i)\}_{i\geq 1})$, see Eqs. (2), (3), (5), and (6). Define also $\tilde{\mathbf{X}}^{[K,M]}(i) = \left(\mathbf{X}_{1,K-1}, \sum_{k=K}^{M} X_k(i), \mathbf{X}_{M+1,N}(i)\right)$, for all $i \geq 1$, and $\{\tilde{\mathbf{D}}^{[K,M]}(i)\}_{i\geq 1} = \phi\left(\left\{\tilde{\mathbf{X}}^{[K,M]}(i)\right\}_{i\geq 1}\right)$. Then, Proposition 1 yields that $\{\tilde{\mathbf{D}}^{[K,M]}(i)\}_{i\geq 1} \leq \{\mathbf{D}(i)\}_{i\geq 1}$. It is clear that ϕ is a non-decreasing functional. Hence, by Lemma 3 and inequality (7), we have $\{\mathbf{D}^{[K,M]}(i)\}_{i\geq 1} \leq_{st} \{\tilde{\mathbf{D}}^{[K,M]}(i)\}_{i\geq 1}$, and the result follows. □

We defer the proofs of Proposition 3 and Corollary 1 as they are based on Proposition 7. We need the following lemma to prove Proposition 4.

Lemma 4 *If a_i and b_i are some positive real numbers for $i = 1, \ldots, n$, where n is a positive integer, then we have*

$$\min_{i=1,\ldots,n} \left\{\frac{a_i}{b_i}\right\} \leq \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}. \tag{16}$$

Proof of Lemma 4 Let $J \in \{1, \ldots, n\}$ be the argument that achieves the minimum in (16), so that $a_J b_i \leq a_i b_J$ for all $i = 1, \ldots, n$. Then, we have

$$b_J \sum_{i=1}^n a_i - a_J \sum_{i=1}^n b_i = \sum_{i=1}^n (a_i b_J - a_J b_i) \geq 0.$$

□

Proof of Proposition 4 Let $T_\infty^{[K,M]}$ be the throughput of the tandem line where stations K through M are parallel pooled and all buffers in the system are replaced by infinite capacity buffers. Then, due to the monotonicity of the throughput of a tandem line in the buffer sizes (see, for example, page 186 in Buzacott and Shanthikumar [8]), we have $T^{[K,M]} \leq T_\infty^{[K,M]}$. We will next show that $T_\infty^{[K,M]} \leq T^{[1,N]}$, which will complete the proof.

Under the assumptions on service times and Assumptions 1 and 2, $T_\infty^{[K,M]}$ exists and satisfies

$$\begin{aligned} T_\infty^{[K,M]} &= \min\left\{\min_{j\in\{1,\ldots,K-1,M+1,\ldots,N\}} \left\{\frac{1}{E[X_j(1)]}\right\}, \sum_{\ell=K}^{M} \theta_\ell \Big/ \sum_{j=K}^{M} \theta_j E[X_j(1)]\right\} \\ &\leq \frac{\sum_{\ell=1}^N \theta_\ell}{\sum_{j=1}^N \theta_j E[X_j(1)]} = T^{[1,N]}, \end{aligned}$$

where the inequality follows from Lemma 4. □

Proof of Proposition 5 Because the service times are i.i.d. and the buffers are infinite, the throughput of the original line is given by $T = 1/\mathrm{E}[X_J(1)]$. Similarly, the throughput of the pooled line where stations K through M are pooled will be determined by the bottleneck station, i.e.,

$$T^{[K,M]} = \min\left\{\min_{j\in\{1,\ldots,N\}\setminus\{K,\ldots,M\}} \left\{\frac{1}{\mathrm{E}[X_j(1)]}\right\}, \frac{M-K+1}{\sum_{j=K}^M \mathrm{E}\left[X_j(1)\right]}\right\},$$

under Assumption 1 and the condition that servers K through M are identical. Hence, if $K \le J \le M$ and $\mathrm{E}[X_j(1)]/\mathrm{E}[X_J(1)] \to 0$ for all $j \in \{1, \dots, N\}\backslash\{J\}$,

$$\frac{T^{[K,M]}}{T} \to M - K + 1.$$

□

Example 2 Suppose that we pool stations 1 and 2 in a tandem line of three stations where $b_1 = \infty$ and $b_2 = b_3 = 0$. Suppose also that the service times at the pooled station satisfy $X_\ell^{[1,2]}(i) = X_1(i) + X_2(i)$ for $\ell = 1, 2$ and $i \ge 1$, $P^{[1,2]} = \infty$, and $Q^{[1,2]} = 0$. For the original line, consider a sample path where $(X_0(1), X_0(2)) = (0, 1)$, $(X_1(1), X_1(2)) = (1, 1)$, $(X_2(1), X_2(2)) = (3, 1)$, and $(X_3(1), X_3(2)) = (1, 3)$ minutes. For the pooled line, suppose that $(X_0^{[1,2]}(1), X_0^{[1,2]}(2)) = (0, 1)$ and $(X_3^{[1,2]}(1), X_3^{[1,2]}(2)) = (1, 2)$ minutes. Note that this example satisfies all conditions of Proposition 1 and the condition that $X_0^{[K,M]}(i) \le X_0(i)$ for all $i \ge 1$, and hence $D_3^{[1,2]}(i) \le D_3(i)$ for $i = 1, 2$. However, in the pooled line, the first job to arrive at the system departs as the second job from the system. This results in a longer sojourn time for this job by pooling. In particular, the sojourn time of the first job arriving to the original line is five minutes, whereas its sojourn time in the pooled line is six minutes.

Proof of Proposition 6 For all $t \ge 0$, let $B_j^{[K,M]}(t)$ be the total number of departures from station $j \in \{1, \dots, K-1, M, \dots, N\}$ by time t in the pooled system and $B_j(t)$ be the total number of departures from station $j \in \{1, \dots, N\}$ by time t in the unpooled system. Let also $B_0^{[K,M]}(t) = B_0(t)$ be the total number of arrivals by time $t \ge 0$ and $D_0^{[K,M]}(i) = D_0(i)$ be the arrival time of job $i \ge 1$ at each system. For notational convenience, assume that $D_K^{[K,M]}(i) = D_M^{[K,M]}(i)$ and $B_K^{[K,M]}(t) = B_M^{[K,M]}(t)$, for all $i \ge 1$ and $t \ge 0$. Then, for all $t \ge 0$, we have

$$H(t) = \sum_{j=0}^{N-1} h_{j+1} \sum_{i=1}^{B_j(t)} \Big(\min\{t, D_{j+1}(i)\} - D_j(i) \Big) \text{ and}$$

$$\begin{aligned} H^{[K,M]}(t) = & \sum_{j \in \{0,\dots,K-2\} \bigcup \{M,\dots,N-1\}} h_{j+1} \sum_{i=1}^{B_j^{[K,M]}(t)} \Big(\min\{t, D_{j+1}^{[K,M]}(i)\} - D_j^{[K,M]}(i) \Big) \\ & + h^{[K,M]} \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \Big(\min\{t, D_K^{[K,M]}(i)\} - D_{K-1}^{[K,M]}(i) \Big). \end{aligned}$$

Consequently, for all $t \ge 0$, we obtain

$$\begin{aligned} & H(t) - H^{[K,M]}(t) \\ & = \sum_{j=K-1}^{N-1} h_{j+1} \sum_{i=1}^{B_j(t)} \Big(\min\{t, D_{j+1}(i)\} - D_j(i) \Big) \end{aligned}$$

$$
\begin{aligned}
&- \sum_{j=M}^{N-1} h_{j+1} \sum_{i=1}^{B_j^{[K,M]}(t)} \left(\min\{t, D_{j+1}^{[K,M]}(i)\} - D_j^{[K,M]}(i)\right) \\
&- h^{[K,M]} \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \left(\min\{t, D_K^{[K,M]}(i)\} - D_{K-1}^{[K,M]}(i)\right) \\
&+ \sum_{j=0}^{K-2} h_{j+1} \left(\sum_{i=1}^{B_j(t)} \left(\min\{t, D_{j+1}(i)\} - D_j(i)\right) \right. \\
&\left. - \sum_{i=1}^{B_j^{[K,M]}(t)} \left(\min\{t, D_{j+1}^{[K,M]}(i)\} - D_j^{[K,M]}(i)\right)\right). \qquad (17)
\end{aligned}
$$

We start by dealing with the sum of the first three terms of Eq. (17). First, note that for all $\ell, m \in \{0, \dots, N-1\}$, $\ell \leq m$, and $t \geq 0$, we have

$$
\begin{aligned}
&\sum_{j=l}^{m} \sum_{i=1}^{B_j(t)} \left(\min\{t, D_{j+1}(i)\} - D_j(i)\right) \\
&= \sum_{j=\ell}^{m} \sum_{i=1}^{B_{j+1}(t)} D_{j+1}(i) + \sum_{j=\ell}^{m} \sum_{i=B_{j+1}(t)+1}^{B_j(t)} t - \sum_{j=\ell}^{m} \sum_{i=1}^{B_j(t)} D_j(i) \\
&= \sum_{i=1}^{B_{m+1}(t)} D_{m+1}(i) + \sum_{i=B_{m+1}(t)+1}^{B_\ell(t)} t - \sum_{i=1}^{B_\ell(t)} D_\ell(i) \\
&= \sum_{i=1}^{B_\ell(t)} \left(\min\{t, D_{m+1}(i)\} - D_\ell(i)\right). \qquad (18)
\end{aligned}
$$

Similarly, for all $\ell, m \in \{0, \dots, K-1\} \cup \{M, \dots, N-1\}$, $\ell \leq m$, and $t \geq 0$, we can obtain

$$
\begin{aligned}
&\sum_{\substack{j \in \{\ell,\dots,m\} \setminus \\ \{K,\dots,M-1\}}} \sum_{i=1}^{B_j^{[K,M]}(t)} \left(\min\{t, D_{j+1}^{[K,M]}(i)\} - D_j^{[K,M]}(i)\right) \\
&\quad = \sum_{i=1}^{B_\ell^{[K,M]}(t)} \left(\min\{t, D_{m+1}^{[K,M]}(i)\} - D_\ell^{[K,M]}(i)\right). \qquad (19)
\end{aligned}
$$

Now, by conditions (iii), (iv), and (v), and Eqs. (18) and (19), the sum of the first three terms of Eq. (17) is greater than or equal to

$$h_K \left\{ \sum_{i=1}^{B_{K-1}(t)} \Big(\min\{t, D_N(i)\} - D_{K-1}(i) \Big) - \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \Big(\min\{t, D_N^{[K,M]}(i)\} - D_{K-1}^{[K,M]}(i) \Big) \right\}. \tag{20}$$

Next, suppose that condition $(ii)(a)$ holds. Then, the fourth term of Eq. (17) reduces to

$$h_K \sum_{i=1}^{B_0(t)} \Big(\min\{t, D_{K-1}(i)\} - \min\{t, D_{K-1}^{[K,M]}(i)\} \Big),$$

by Eqs. (18) and (19). Then, using Eq. (20), we have

$$\begin{aligned}
H(t) - H^{[K,M]}(t) \geq h_K & \left\{ \sum_{i=1}^{B_{K-1}(t)} \Big(\min\{t, D_N(i)\} - D_{K-1}(i) \Big) \right. \\
& - \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \Big(\min\{t, D_N^{[K,M]}(i)\} - D_{K-1}^{[K,M]}(i) \Big) + \sum_{i=1}^{B_0(t)} \min\{t, D_{K-1}(i)\} \\
& \left. - \sum_{i=1}^{B_0(t)} \min\{t, D_{K-1}^{[K,M]}(i)\} \right\} \\
= h_K & \left\{ \sum_{i=1}^{B_{K-1}(t)} \min\{t, D_N(i)\} - \sum_{i=1}^{B_{K-1}(t)} D_{K-1}(i) - \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \min\{t, D_N^{[K,M]}(i)\} \right. \\
& + \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} D_{K-1}^{[K,M]}(i) + \sum_{i=1}^{B_{K-1}(t)} D_{K-1}(i) + \sum_{i=B_{K-1}(t)+1}^{B_0(t)} t \\
& \left. - \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} D_{K-1}^{[K,M]}(i) - \sum_{i=B_{K-1}^{[K,M]}(t)+1}^{B_0(t)} t \right\} \\
= h_K & \left\{ \sum_{i=1}^{B_{K-1}(t)} \min\{t, D_N(i)\} - \sum_{i=1}^{B_{K-1}^{[K,M]}(t)} \min\{t, D_N^{[K,M]}(i)\} + \sum_{i=B_{K-1}(t)+1}^{B_0(t)} t \right. \\
& \left. - \sum_{i=B_{K-1}^{[K,M]}(t)+1}^{B_0(t)} t \right\} \\
= h_K & \sum_{i=1}^{B_0(t)} \Big(\min\{t, D_N(i)\} - \min\{t, D_N^{[K,M]}(i)\} \Big),
\end{aligned}$$

which is nonnegative by condition (i).

Finally, suppose that condition $(ii)(b)$ holds, in which case we have $D_j(i) = D_j^{[K,M]}(i)$ and $B_j(t) = B_j^{[K,M]}(t)$ for $j \in \{0, \ldots, K-1\}$, $i \geq 1$, and $t \geq 0$. Then, the fourth term of Eq. (17) becomes zero, and using Eq. (20) and condition (i), we have

$$H(t) - H^{[K,M]}(t) \geq h_K \sum_{i=1}^{B_{K-1}(t)} \left(\min\{t, D_N(i)\} - \min\{t, D_N^{[K,M]}(i)\}\right) \geq 0.$$

□

Proof of Proposition 7 Under Assumptions 2 and 3, we have

$$T^{(1,N)} = \lim_{n\to\infty} \frac{n}{\sum_{i=1}^n X^{(1,N)}(i)} = \lim_{n\to\infty} \frac{n\sum_{\ell=1}^N \theta_\ell}{\sum_{i=1}^n \sum_{j=1}^N \theta_j X_j(i)},$$

and, under Assumptions 1 and 2, we have

$$T^{[1,N]} = \sum_{\ell=1}^N \lim_{n\to\infty} \frac{n}{\sum_{i=1}^n X_\ell^{[1,N]}(i)} = \lim_{n\to\infty} \frac{n\sum_{\ell=1}^N \theta_\ell}{\sum_{i=1}^n \sum_{j=1}^N \theta_j X_j(i)}.$$

These limits exist and are equal by the strong law of large numbers because $\left\{\sum_{j=1}^N \theta_j X_j(i)\right\}_{i\geq 1}$ is an i.i.d. sequence of random variables with finite mean, which completes the proof. □

Proof of Proposition 3 Because $\{\mathbf{X}(i)\}_{i\geq 1}$ is a sequence of i.i.d. random vectors with finite component means and $\theta_j \in [0, \infty)$ for all $j \in \{1, \ldots, N\}$, $\{\sum_{j=1}^N \theta_j X_j(i)\}_{i\geq 1}$ is a sequence of i.i.d. random variables with finite mean. Hence, Proposition 7 yields that $T^{[1,N]} = T^{(1,N)}$ if Assumptions 1, 2, and 3 hold. Combining this with the fact that $T^{(1,N)} \geq T$ under Assumption 3 by Theorem 1 in Argon and Andradóttir [4] completes the proof. □

Proof of Corollary 1 Let $t_{m,n}$ denote the throughput of the tandem line that is obtained by removing stations 1 through $m-1$ and stations $n+1$ through N in the original line, where $1 \leq m \leq n \leq N$. If in the original line $b_K = b_{M+1} = \infty$, then its throughput will exist and be equal to $\min\{t_{1,K-1}, t_{K,M}, t_{M+1,N}\}$ (see, for example, Muth [19]) under the assumption that the service times are i.i.d. with finite mean. Moreover, since the throughput of a tandem line decreases with a decrease in the buffer sizes (see, for example, page 186 in Buzacott and Shanthikumar [8]), we have

$$T \leq \min\{t_{1,K-1}, t_{K,M}, t_{M+1,N}\} \tag{21}$$

as b_K and b_{M+1} are not necessarily infinite in the original line.

Now, let $t^{[K,M]}$ be the throughput of the system that consists of only the pooled station with an infinite supply of jobs in front of the pooled station and infinite room following it. If the buffers before and after the pooled station are infinite, then we have $T^{[K,M]} = \min\{t_{1,K-1}, t^{[K,M]}, t_{M+1,N}\}$. Using Proposition 3, which implies that $t^{[K,M]} \geq t_{K,M}$, and inequality (21), we have $T \leq T^{[K,M]}$. (Note that we here use the fact that Proposition 3 is still valid assuming that there is a stochastic arrival stream at the first station and $b_1 = \infty$. See Sect. 3 for this result.) □

Proof of Corollary 2 Let $t^{(K,M)}$ be the throughput of the system that consists of only the pooled station under cooperative pooling with an infinite supply of jobs in front of the pooled station and infinite room following it. If the buffers before and after the pooled stations are infinite, then we have $T^{[K,M]} = \min\{t_{1,K-1}, t^{[K,M]}, t_{M+1,N}\}$ and $T^{(K,M)} = \min\{t_{1,K-1}, t^{(K,M)}, t_{M+1,N}\}$ under the given assumption on service times. (See the proof of Corollary 1 for definitions of $t_{1,K-1}$ and $t_{M+1,N}$.) Now, using Proposition 7, we have $t^{[K,M]} = t^{(K,M)}$, which implies that $T^{(K,M)} = T^{[K,M]}$. (Note that we here use the fact that Proposition 7 is still valid assuming that there is a stochastic arrival stream at the first station and $b_1 = \infty$. See Sect. 3 for this result.) □

Proof of Proposition 8 Each one of the four systems can be modeled as a birth-death process. We start with System 0; the others can be derived from this birth–death model by simple substitution. Let the system state be the number of jobs that have finished service at station 1 but not at station 2. Then, the state space will be given by $\mathcal{S} = \{0, 1, \ldots, L_1+L_2+B_0\}$. Let $\lambda(i)$ be the birth rate in state i for $i = 0, 1, \ldots, L_1+L_2+B_0-1$ and $\theta(i)$ be the death rate in state i for $i = 1, 2, \ldots, L_1+L_2+B_0$. We have:

$$\lambda(i) = \begin{cases} L_1\mu_1, & \text{for } i = 0, \ldots, L_2+B_0, \\ (L_1+L_2+B_0-i)\mu_1, & \text{for } i = L_2+B_0+1, \ldots, L_1+L_2+B_0-1; \end{cases}$$

$$\theta(i) = \begin{cases} i\mu_2, & \text{for } i = 1, \ldots, L_2-1, \\ L_2\mu_2, & \text{for } i = L_2, \ldots, L_1+L_2+B_0. \end{cases}$$

Next, we let $\pi(i)$ be the limiting probability of being in state $i \in \mathcal{S}$. Note that the limiting distribution for this birth–death process exists (because the state space is finite) and is given by $\pi(i) = f(i)\alpha^i\pi(0)$ for $i \in \mathcal{S}$, where $\alpha = L_1\mu_1/(L_2\mu_2)$,

$$f(i) = \begin{cases} \frac{L_2^i}{i!}, & \text{for } i = 0, \ldots, L_2-1, \\ \frac{L_2^{L_2}}{L_2!}, & \text{for } i = L_2, \ldots, L_2+B_0+1, \\ \frac{L_2^{L_2} L_1^{L_2+B_0-i} L_1!}{L_2!(L_1+L_2+B_0-i)!}, & \text{for } i = L_2+B_0+2, \ldots, L_1+L_2+B_0, \end{cases}$$

and

$$\pi(0) = \left(\sum_{i=0}^{L_1+L_2+B_0} f(i)\alpha^i \right)^{-1}.$$

Then, the steady-state throughput of System 0 is given by $T_0 = \pi(0) \sum_{i=1}^{L_1+L_2+B_0} \theta(i) f(i)\alpha^i$.

To obtain the steady-state throughput for System j (for $j = 1, 2, 3$), replace B_0 with B_j in the above expressions for System 0. Furthermore, for System j, where $j = 1, 2$, replace L_j and μ_j with 1 and $L_j\mu_j$, respectively. Finally, for System 3, replace L_i and μ_i with 1 and $L_i\mu_i$, respectively, for $i = 1, 2$. The steady-state throughputs are then given as follows:

$$T_0 = L_1\mu_1 \left(\frac{\sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^i}{i!} + \frac{L_2^{L_2}\alpha^{L_2}}{L_2!} \sum_{i=0}^{B_0-1} \alpha^i + \frac{L_2^{L_2} L_1!\alpha^{L_1+L_2+B_0-1}}{L_1^{L_1} L_2!} \sum_{i=0}^{L_1-1} \frac{\alpha^{-i} L_1^i}{i!}}{\sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^i}{i!} + \frac{L_2^{L_2}\alpha^{L_2}}{L_2!} \sum_{i=0}^{B_0-1} \alpha^i + \frac{L_2^{L_2} L_1!\alpha^{L_1+L_2+B_0}}{L_1^{L_1} L_2!} \sum_{i=0}^{L_1} \frac{\alpha^{-i} L_1^i}{i!}} \right), \tag{22}$$

$$T_1 = L_1\mu_1 \left(\frac{\sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^i}{i!} + \frac{L_2^{L_2}\alpha^{L_2}}{L_2!} \sum_{i=0}^{B_1} \alpha^i}{\sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^i}{i!} + \frac{L_2^{L_2}\alpha^{L_2}}{L_2!} \sum_{i=0}^{B_1+1} \alpha^i} \right), \tag{23}$$

$$T_2 = L_1\mu_1 \left(\frac{\sum_{i=0}^{B_2} \alpha^i + \frac{L_1!\alpha^{L_1+B_2}}{L_1^{L_1}} \sum_{i=0}^{L_1-1} \frac{\alpha^{-i} L_1^i}{i!}}{\sum_{i=0}^{B_2} \alpha^i + \frac{L_1!\alpha^{L_1+B_2+1}}{L_1^{L_1}} \sum_{i=0}^{L_1} \frac{\alpha^{-i} L_1^i}{i!}} \right), \tag{24}$$

$$T_3 = L_1\mu_1 \left(\frac{\sum_{i=0}^{B_3+1} \alpha^i}{\sum_{i=0}^{B_3+2} \alpha^i} \right). \tag{25}$$

We next perform a pairwise comparison of the steady-state throughputs of these four systems.

System 0 versus System 1: From Eqs. (22) and (23), we find that $T_0 \le T_1$ if and only if

$$\left(\sum_{i=0}^{B_1} \alpha^i - \sum_{i=0}^{B_0} \alpha^i - \frac{L_1!\alpha^{L_1+B_0-1}}{L_1^{L_1}} \sum_{i=0}^{L_1-2} \frac{\alpha^{-i} L_1^i}{i!} \right) \left(\sum_{i=0}^{L_2} \frac{L_2^i\alpha^i}{i!} - \sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^{i+1}}{i!} \right) \ge 0. \tag{26}$$

The term in the second parentheses above reduces to

$$1 + \sum_{i=0}^{L_2-1} \frac{L_2^i\alpha^{i+1}}{(i+1)!}(L_2 - 1 - i),$$

which is greater than zero. Hence, $T_0 \le T_1$ if and only if the term in the first parentheses in (26) is nonnegative.

We first consider the case where $B_1 = B_0 + L_1 - 1$, so that System 1 has the same number of spaces for jobs as System 0. For this case, the term in the first parentheses in (26) reduces to

$$\frac{L_1!\alpha^{B_0+1}}{L_1^{L_1}}\sum_{i=0}^{L_1-2}\alpha^{L_1-2-i}\left(\frac{L_1^{L_1}}{L_1!}-\frac{L_1^i}{i!}\right),$$

which is greater than zero because $L_1^{L_1-i}i! > L_1!$ for all $i = 0, 1, \ldots, L_1 - 2$. Thus, when $B_1 = B_0 + L_1 - 1$, we have $T_0 < T_1$.

We next consider the case where $B_1 = B_0$, so that Systems 0 and 1 have the same number of buffer spaces excluding the spaces for servers. For this case, the term in the first parentheses in (26) reduces to $-L_1!\sum_{i=0}^{L_1-2}\alpha^{L_1+B_0-1-i}L_1^{i-L_1}/i! < 0$. Thus, when $B_1 = B_0$, we have $T_0 > T_1$.

System 0 versus System 2: From Eqs. (22) and (24), we find that $T_0 \le T_2$ if and only if

$$\left(\frac{L_2^{L_2}\alpha^{B_0+L_2-B_2-1}}{L_2!}\sum_{i=0}^{B_2}\alpha^i-\sum_{i=0}^{L_2-1}\frac{L_2^i\alpha^i}{i!}-\frac{L_2^{L_2}\alpha^{L_2}}{L_2!}\sum_{i=0}^{B_0-1}\alpha^i\right)$$
$$\left(\frac{L_1^{L_1}\alpha^{1-L_1}}{L_1!}-(1-\alpha)\sum_{i=0}^{L_1-1}\frac{\alpha^{-i}L_1^i}{i!}\right)\ge 0. \tag{27}$$

We can show that the term in the second parentheses above is positive as follows:

$$\begin{aligned}\frac{L_1^{L_1}\alpha^{1-L_1}}{L_1!}-(1-\alpha)\sum_{i=0}^{L_1-1}\frac{\alpha^{-i}L_1^i}{i!}&=\sum_{i=0}^{L_1}\frac{\alpha^{1-i}L_1^i}{i!}-\sum_{i=0}^{L_1-1}\frac{\alpha^{-i}L_1^i}{i!}\\&=\alpha+\sum_{i=0}^{L_1-1}\frac{L_1^i\alpha^{-i}}{(i+1)!}(L_1-1-i)>0.\end{aligned}$$

Hence, $T_0 \le T_2$ if and only if the term in the first parentheses in (27) is nonnegative.

We first consider the case where $B_2 = B_0 + L_2 - 1$, so that System 2 has the same number of spaces for jobs as System 0. For this case, the term in the first parentheses in (27) reduces to

$$\sum_{i=0}^{L_2-2}\left(\frac{L_2^{L_2}}{L_2!}-\frac{L_2^i}{i!}\right)\alpha^i, \tag{28}$$

which is positive because $L_2^{L_2-i}i! > L_2!$ for all $i = 0, 1, \ldots, L_2 - 2$. Thus, when $B_2 = B_0 + L_2 - 1$, we have $T_0 < T_2$.

We next consider the case where $B_2 = B_0$, so that Systems 0 and 2 have the same number of buffer spaces excluding the spaces for servers. For this case, the term in the first parentheses in (27) reduces to $-\sum_{i=0}^{L_2-2}L_2^i\alpha^i/i! < 0$. Thus, when $B_2 = B_0$, we have $T_0 > T_2$.

System 1 versus System 3: From Eqs. (23) and (25), we find that $T_1 \le T_3$ if and only if

$$\sum_{i=0}^{L_2-1}\frac{L_2^i\alpha^i}{i!}+\frac{L_2^{L_2}\alpha^{L_2}}{L_2!}\left(\sum_{i=0}^{B_1}\alpha^i-\sum_{i=0}^{B_3+1}\alpha^{i+B_1-B_3-1}\right)\leq 0. \tag{29}$$

We first consider the case where $B_3 = B_1 + L_2 - 1$, so that System 3 has the same number of spaces for jobs as System 1. For this case, the left-hand side of (29) reduces to (28) multiplied by negative one, which is less than zero. Thus, when $B_3 = B_1+L_2-1$, we have $T_1 < T_3$.

We next consider the case where $B_3 = B_1$, so that Systems 1 and 3 have the same number of buffer spaces excluding the spaces for servers. For this case, the left-hand side of (29) reduces to $\sum_{i=0}^{L_2-2}\alpha^i L_2^i/i!$, which is positive. Hence, in this case, we have $T_1 > T_3$.

System 2 versus System 3: From Eqs. (24) and (25), we find that $T_2 \leq T_3$ if and only if

$$\sum_{i=0}^{B_2}\alpha^{i+B_3+1-B_2}+\frac{L_1!\alpha^{L_1-1}}{L_1^{L_1}}\sum_{i=0}^{L_1-1}\frac{L_1^i\alpha^{-i}}{i!}-\sum_{i=0}^{B_3+1}\alpha^i\leq 0. \tag{30}$$

We first consider the case where $B_3 = B_2 + L_1 - 1$, so that System 3 has the same number of spaces for jobs as System 2. For this case, the left-hand side of (30) reduces to $\sum_{i=0}^{L_1-2}(L_1!/(L_1^{L_1-i}i!)-1)\alpha^{L_1-1-i}$, which is less than zero because $L_1^{L_1-i}i! > L_1!$ for all $i = 0, 1, \ldots, L_1 - 2$. Thus, when $B_3 = B_2 + L_1 - 1$, we have $T_2 < T_3$.

We next consider the case where $B_3 = B_2$, so that Systems 2 and 3 have the same number of buffer spaces excluding the spaces for servers. For this case, the left-hand side of (30) reduces to $L_1!\sum_{i=0}^{L_1-2}L_1^{i-L_1}\alpha^{L_1-1-i}/i! > 0$. Hence, in this case, we have $T_2 > T_3$.

In order to complete the proof, we need to show that T_0 is a non-decreasing function of B_0, which implies that T_j is a non-decreasing function of B_j for $j = 1, 2, 3$. (Our literature search failed to find the exact monotonicity result in published work. For a similar monotonicity result in the case of tandem lines with two or more stations each having a single server, see [18].) From Eq. (22), we find that T_0 with $B_0 + 1$ buffers is greater than equal to T_0 with B_0 buffers if and only if

$$\left(\sum_{i=0}^{L_1-1}\frac{\alpha^{L_1-i}L_1^i}{i!}-\sum_{i=0}^{L_1-2}\frac{\alpha^{L_1-1-i}L_1^i}{i!}\right)\left(\sum_{i=0}^{L_2}\frac{L_2^i\alpha^i}{i!}-\sum_{i=0}^{L_2-1}\frac{L_2^i\alpha^{i+1}}{i!}\right)\geq 0. \tag{31}$$

We know from the argument following (26) that the term in the second parentheses above is positive. Furthermore, the term in the first parentheses in (31) reduces to

$$\alpha^{L_1}+\sum_{i=1}^{L_1-1}\frac{\alpha^{L_1-i}L_1^{i-1}}{i!}(L_1-i),$$

which is greater than zero. This concludes the proof. $\square$

References

1. Andradóttir, S., Ayhan, H., Down, D.G.: Server assignment policies for maximizing the steady-state throughput of finite queueing systems. Manag. Sci. **47**(10), 1421–1439 (2001)
2. Andradóttir, S., Ayhan, H., Down, D.G.: Dynamic server allocation for queueing networks with flexible servers. Oper. Res. **51**(6), 952–968 (2003)
3. Andradóttir, S., Ayhan, H., Down, D.G.: Optimal assignment of servers to tasks when collaboration is inefficient. Queueing Syst. **75**(1), 79–110 (2013)
4. Argon, N.T., Andradóttir, S.: Partial pooling in tandem lines with cooperation and blocking. Queueing Syst. **52**(1), 5–30 (2006)
5. Bartholdi III, J.J., Eisenstein, D.D., Foley, R.D.: Performance of bucket brigades when work is stochastic. Operations Res, **49**(5), 710–719 (2001)
6. Benjaafar, S.: Performance bounds for the effectiveness of pooling in multi-processing systems. Eur. J. Oper. Res. **87**(2), 375–388 (1995)
7. Buzacott, J.A.: Commonalities in reengineered business processes: models and issues. Manag. Sci. **42**(5), 768–782 (1996)
8. Buzacott, J.A., Shanthikumar, J.G.: Stochastic Models of Manufacturing Systems. Prentice Hall, Englewood Cliffs (1993)
9. Calabrese, J.B.: Optimal workload allocation in open networks of multiserver queues. Manag. Sci. **38**(12), 1792–1802 (1992)
10. Glasserman, P., Yao, D.D.: A GSMP framework for the analysis of production lines. In: Yao, D.D. (ed.) Chapter 4 Stochastic Modeling and Analysis of Manufacturing Systems, pp. 133–188. Springer, New York (1994)
11. Glasserman, P., Yao, D.D.: Structured buffer-allocation problems. Discrete Event Dyn. Syst. Theory Appl. **6**(1), 9–41 (1996)
12. Harel, A.: Convexity results for the Erlang delay and loss formulae when the server utilization is held constant. Oper. Res. **59**(6), 1420–1426 (2011)
13. Hillier, F.S., So, K.C.: On the simultaneous optimization of server and work allocations in production line systems with variable processing times. Oper. Res. **44**(3), 435–443 (1996)
14. Hillier, F.S., So, K.C., Boling, R.W.: Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times. Manag. Sci. **39**(1), 126–133 (1993)
15. Hopp, W.J., Tekin, E., Van Oyen, M.P.: Benefits of skill chaining in serial production lines with cross-trained workers. Manag. Sci. **50**(1), 83–98 (2004)
16. Kulkarni, V.G.: Modeling and Analysis of Stochastic Systems, 2nd edn. CRC Press, Boca Raton (2010)
17. Mandelbaum, A., Reiman, M.I.: On pooling in queueing networks. Manag. Sci. **44**(7), 971–981 (1998)
18. Meester, L.E., Shanthikumar, J.G.: Concavity of the throughput of tandem queueing systems with finite buffer storage space. Adv. Appl. Probab. **22**(3), 764–767 (1990)
19. Muth, E.J.: The production rate of a series of work stations with variable service times. Int. J. Prod. Res. **11**(2), 155–169 (1973)
20. Shaked, M., Shanthikumar, J.G.: Stochastic Orders. Springer, New York (2007)
21. Shanthikumar, J.G., Yao, D.D.: Second-order stochastic properties in queueing systems. Proc. IEEE **77**(1), 162–170 (1989)
22. Smith, D.R., Whitt, W.: Resource sharing for efficiency in traffic systems. Bell Syst. Tech. J. **60**(1), 39–55 (1981)
23. So, K.C.: Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines. IIE Trans. **29**(1), 81–88 (1997)
24. Tekin, E., Hopp, W.J., Van Oyen, M.P.: Pooling strategies for call center agent cross-training. IIE Trans. **41**(6), 546–561 (2009)
25. Van Oyen, M.P., Gel, E.G.S., Hopp, W.J.: Performance opportunity of workforce agility in collaborative and noncollaborative work systems. IIE Trans. **33**(9), 761–777 (2001)
26. Yamazaki, G., Sakasegawa, H., Shanthikumar, G.: On optimal arrangement of stations in a tandem queueing system with blocking. Manag. Sci. **38**(1), 137–153 (1992)