Serial Agile Production Systems with Automation

Wallace J. Hopp, Seyed M. R. Iravani, Biying Shou
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208
{hopp@northwestern.edu, s-iravani@northwestern.edu, biying@iems.northwestern.edu}

To gain insights into the design and control of manufacturing cells with automation, we study simple models of serial production systems where one flexible worker attends a set of automated stations. We (a) characterize the operational benefits of automation, (b) determine the most desirable placement of automation within a line, and (c) investigate how best to allocate labor dynamically in a line with manual and automatic equipment. We do this by first considering two-station Markov decision process models and then studying three-station simulations. Our results show that the capacity of production lines with automatic machines can be significantly lower than the rate of the bottleneck. We also show that automating a manual machine can have a dramatic effect on the average work-in-process (WIP) level, provided that labor is the system bottleneck. Once a machine becomes the bottleneck, the benefits from further automation are dramatically reduced. In general, we find that automation is more effective when placed toward the end of the line rather than toward the front. Finally, we show that automation level increases the priority workers should give to a station when selecting a work location.

Subject classifications: cross-trained workers; serial line; Markov decision process; automation; capacity.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received August 2002; revision received October 2004; accepted October 2004.

1. Introduction

Revolutionary changes in information technology, globalization of markets, and competition have radically altered manufacturing systems over the past two decades. Under pressure to continually improve the price, variety, and responsiveness they offer to customers, firms have increasingly moved toward highly flexible production facilities making use of automated flexible machinery and cross-trained workers. We refer to the emerging manufacturing paradigm that relies on these two elements as agile automated production (AAP). While there is ample evidence that U.S. manufacturers are adopting AAP, there is still needed during the entire processing time. For our purpose, these are considered manual operations because the worker's presence is still needed during the entire processing time.

To address the above issues, we start with a two-station line and we analyze the relationship between the capacity of the line and the rate of the bottleneck. We then formulate a Markov decision process (MDP) to find the basic operations in the following sequence: (i) loading, (ii) setup, (iii) machine processing, and (iv) unloading. In general, the worker must be present for steps (i), (ii), and (iv), but not during step (iii), which is automated. We refer to production environments with this type of equipment as machine-based environments, and we refer to the corresponding type of job-processing operations as automatic operations. Note that there exist some cases where the worker is also needed to supervise the automatic operation to prevent the accumulation of waste material, or check the quality of the cut, etc. For our purpose, these are considered manual operations because the worker's presence is still needed during the entire processing time.

In this paper, we study the design and control of serial lines with a single cross-trained worker in a machine-based environment that consists of a combination of manual and automatic machines. In this context, we consider the following questions:

1. What factors affect the line's capacity?
2. When is automation most attractive for improving operational efficiency?
3. Where in the line is automation most effective?
4. Is concentrated (focused on one machine) or distributed (spread over several machines) automation more effective?
5. How is performance of a line with automation affected by the worker's operating policy?

To address the above issues, we start with a two-station line and we analyze the relationship between the capacity of the line and the rate of the bottleneck. We then formulate a Markov decision process (MDP) to find the
optimal dynamic assignment policy for the worker. This MDP enables us to characterize the optimal operating policy, evaluate the benefits of automating a single machine, and examine the effects of the position of the automated machine. However, the optimal policy is too complex to extend to longer lines or to use in practical settings. Therefore, we turn our attention to two simple policies that are representative of how such systems are staffed in practice: (i) fixed-priority policies, in which the worker chooses a machine to work at according to a static priority rule; and (ii) cyclic policies, in which the server attends machines in a cyclic fashion. We then evaluate the questions of the benefits, placement, and concentration of automation for three-station lines using simulation of these heuristic policies.

2. Literature

Almost all the literature on production systems with cross-trained workers has considered AWP environments. Examples of work on serial production systems with cross-trained workers are Bartholdi and Eisenstein (1996) on bucket brigades; McClain et al. (2000), McClain et al. (1992), Ostolaza et al. (1990), and Zavadlav et al. (1996) on work sharing; and Farrar (1993), Iravani et al. (1997), Duenyas et al. (1998), Van Oyen et al. (2001), Gel et al. (2000), Ahn et al. (1999), and Gupta et al. (1987) on various systems involving flexible labor.

Although manufacturing cells are usually machine-based environments, most of the literature on cellular manufacturing with cross-trained operators (i.e., dual resource cellular manufacturing) model these systems as AWP environments by assuming (explicitly or implicitly) that the operator must supervise the machine or control the operation while it is processing a job. Reviews of this research can be found in Trelaven (1989) and Askin and Strada (1999).

Research into AWP systems has provided many useful insights into worker cross-training policies. Unfortunately, most of these do not extend to AAP systems. Nakade et al. (1997) present industrial examples where the ratio of automated processing time (which does not require operator presence) to total processing time is as high as 0.8. This means that if the operator is cross trained, s/he will have ample opportunity to operate another machine while the automated machine is running. Studies on AWP systems neglect this opportunity because they model all operations as manual.

There has been some work that explicitly models automation in AAP systems. For example, Nakade, Ohno and others (Ohno and Nakade 1997; Nakade and Ohno 1995, 1997; Nakade et al. 1997) analyze a serial AAP system that they call a single-unit production and conveyance system (SPC) (called ikko-nagashi in Japanese; see Monden 1993) consisting of a serial line with one machine in each station and cross-trained machine operators. Each operator is responsible for multiple machines and visits them in cyclic fashion. When the operator arrives at one of the machines, s/he waits for the end of processing of the preceding job if it is not completed, and then unloads the processed job, puts it on a chute to roll to the next machine, loads the new job on the machine, switches the machine on, and then goes to the next machine. They obtain performance measures such as cycle time and worker waiting time under this cyclic policy. Nakade and Ohno (1995, 1997) show the reversibility of this system, that is, that the expected cycle time of the reversed system where each worker operates and walks in the reversed order of stations is the same as that of the original system.

Desruelle and Steudel (1996) investigate a similar system in the context of work cell design. Their model considers identical machines, different part types, and detailed operations such as machine loading and unloading, machine setups, and part processing where machine processing cycles are automatic and do not require manual intervention. By modeling the work cell as two interacting queueing networks—an open part/machine network and a closed machine/operator network—they evaluate machine utilization and waiting times for the operator.

While the above papers provide a useful body of knowledge on AWP systems and a start to modeling AAP systems, they do not address:

- the structure of the optimal operating policy for a cross-trained worker in an AAP system;
- the operational benefits of automation (e.g., reducing cycle time and WIP); and
- the impact of the level, position, and concentration of automation.

This paper is aimed directly at these issues. Using analytical and simulation models, we develop a better understanding of the effects of automation in a single-worker serial agile production system.

The remainder of this paper is organized as follows. In §3, we introduce new concepts for characterizing bottlenecks and capacity in AAP systems. Section 4 analyzes a two-station line model with one automated machine. Using MDPs, we characterize the structure of the optimal policy and examine the impacts of automation level and the position of the automation in the line. In §5, we extend our study to three-station lines with multiple automated machines to see how our observations about automation benefits, level, placement, and concentration hold up in more complex systems. We conclude the paper in §6.

3. Bottleneck and Capacity in Serial AAP Systems

In serial lines, the bottleneck is defined as the resource that has the highest utilization in the system. In traditional serial production lines with workers dedicated to stations and no yield loss or rework, the bottleneck is the station (or stations) with the largest processing time. Furthermore, the capacity of the line, which is defined as the maximum throughput of the system, is equal to the maximum production rate of the bottleneck station. However, in serial AAP
lines in which there are fewer workers than stations, and some or all stations have automated machines, the concept of a bottleneck becomes more complex, as does the capacity of the system. In this section, we define the bottleneck and examine its effect on capacity in serial AAP systems with one fully cross-trained worker.

We begin by introducing the following notation. We call stations with automatic machines, automated stations, and other stations, manual stations. Let $N_a$ and $N_m$ be the sets of automated and manual stations in an $N$-station AAP line, respectively. For each manual station $j$ ($j \in N_m$), we define $t_j$ as the total operation time at that station, and for each automated station $i$ ($i \in N_a$), we define:

$$1/l_i = \text{average job loading time on the machine at station } i,$$

$$1/\mu_i = \text{average job processing time on the machine at station } i,$$

$$1/\alpha_i = \text{average job unloading time on the machine at station } i,$$  

$$t_i = \text{total average operation time required to finish a job at station } i,$$  

which is given by

$$t_i = \frac{1}{l_i} + \frac{1}{\mu_i} + \frac{1}{\alpha_i}, \quad i \in N_a,$$

$\omega_i$ is the amount of time a station requires to perform a job, measured as the average operation time at station $i$ performed automatically by the machine, which is given by $\omega_i = 1/\mu_i$ for all $i \in N_a$.

$\Omega_i = \text{percent of automation at station } i$, measured by the percentage of the entire operation at station $i$ that is automated, which is given by $\Omega_i = \omega_i/l_i$ for all $i \in N_a$.

We define $t_0$ as the total average time of the manual operations required to complete a job, which is

$$t_0 = \sum_{i \in N_m} \left( \frac{1}{l_i} + \frac{1}{\mu_i} \right) + \sum_{j \in N_a} t_j.$$  

Note that $t_0$ is the total average time a worker spends on a job. Therefore, if the job arrival rate to the line is $\lambda$, then the worker utilization $\rho_i$ will be $\rho_i = \lambda t_0$.

We define the bottleneck time $t_b$ in the line as follows:

$$t_b = \max \{ t_0, t_1, t_2, \ldots, t_N \},$$

and therefore,

$$\sum_{j \in N_m} t_j + \sum_{i \in N_a, i \neq k} \left( \frac{1}{l_i} + \frac{1}{\mu_i} \right) < \frac{1}{l_k} + \frac{1}{\mu_k} + \frac{1}{u_k}.$$  

Inequality (1) implies that during the automatic processing time at station $k$, the worker will have enough time, on average, to finish loading and unloading all other automated stations ($i \in N_a$ and $i \neq k$), and also to finish processing a job in all manual stations ($j \in N_m$) in the line. Under these circumstances, even if there is infinite work in progress (WIP) in the line, the line cannot produce more than the capacity of the machine at station $k$ (i.e., $1/t_k$ per unit time). Hence, line capacity is limited by the machine bottleneck rate.

2. Worker bottleneck. When $t_b = t_0 > t_i$ for all $i = 1, 2, \ldots, N$, the total manual work required to finish one job in the line is larger than the time at any station. Therefore, even when the WIP in the system is infinite and the worker is 100% utilized, the line cannot produce more than the capacity of the worker (i.e., $1/t_0$ jobs per unit time). Therefore, the capacity of the line is limited by the worker bottleneck.

3. Machine and worker bottleneck. If $t_b = t_0 = t_k$ for some $k \in N_a$, the worker and at least one of the machines are bottlenecks. In this case, the capacity of the line is limited by the worker as well as the bottleneck machine(s).

In traditional serial production lines, the capacity of the line is equal to the bottleneck rate, $\mu_k = 1/t_b$. That is, when the job arrival rate approaches the bottleneck rate, the utilization of the bottleneck station approaches 100%. Hence, the production rate of the line approaches the bottleneck rate. However, in a serial AAP line, increasing the arrival rate to (or above) the bottleneck rate does not necessarily guarantee 100% utilization of the bottleneck. The reason is that both machines and labor are needed to complete job processing, and hence interference can occur.

To illustrate this phenomenon, we analyze the behavior of a one-worker, two-station AAP system in Lemmas 1 and 2, in which: (i) Loading and unloading times are stochastic (because manual operations are generally subject to human variability), while job-processing times can be either deterministic or stochastic, and (ii) if a manual operation is preempted, the operation can be restarted from where it was preempted (i.e., preempt-resume).

In Lemma 1 we make use of the cyclic policy, so we now describe it in more detail. This policy is commonly used in manufacturing cells with all machines automated and ample raw material (see Nakade and Ohno 1995, Nakade et al. 1997), but can be adapted to our AAP system with a mixture of manual and automatic machines. We illustrate how with a simple three-station line for which there are two cyclic policies, denoted by 1-3-2 and 1-2-3, where the numbers indicate the order in which stations are visited in each cycle. When the worker arrives at a manual station, s/he processes the job in that station before switching to
the next station in the sequence. When the worker arrives at an automated station where the machine has already finished processing a job, s/he unloads the machine, reloads it, and then switches to the next machine. However, if upon the worker’s arrival, the machine in the automated station is still processing a job, the worker waits until the station completes the processing and then unloads and reloads before moving to the next station. It should be apparent that the cyclic policy may result in worker idleness.

Define \(X_{u_1}, X_{l_1},\) and \(X_{\mu_1}\) as the random variables representing unloading, loading, and automatic processing times on machine \(i\), respectively. Define Conditions E1 and E2 as follows:

**Condition E1.** \(\Pr\{X_{u_1} + X_{l_1} < X_{\mu_1}\} = 0.\)

**Condition E2.** \(\Pr\{X_{u_1} + X_{l_1} < X_{\mu_2}\} = 0.\)

Lemmas 1 and 2 analyze the capacity of two-station AAP lines; proofs are given in the online appendix.

**Lemma 1.** In a serial two-station AAP system with automated machines and one fully cross-trained worker, if the worker is the bottleneck \((t_b = t_0)\), then

(i) If both stations are automated and both Conditions E1 and E2 do not hold, the capacity of the line is strictly less than the bottleneck rate \(\mu_b = 1/t_b\).

(ii) If both stations are automated and both Conditions E1 and E2 hold, the capacity of the line is the bottleneck rate \(\mu_b = 1/t_b\). This capacity can be attained under a cyclic policy.

(iii) If only one station is automated, the capacity of the line is the bottleneck rate \(\mu_b = 1/t_b\).

Parts (i) and (ii) of Lemma 1 also hold in preempt-repeat systems in which a preempted operation must be started from the beginning. However, part (iii) of the lemma may not apply in this case. The reason is that in those lines, 100% utilization of the worker does not guarantee a capacity equal to the bottleneck rate \(\mu_b = 1/t_b\).

**Lemma 2.** In a serial two-station AAP system with one fully cross-trained worker:

(i) If both stations are automated and both machines are bottlenecks \((t_b = t_1 = t_2)\), then the capacity of the line is strictly less than the bottleneck rate \(\mu_b = 1/t_i\) \((i = 1 \text{ or } 2)\).

(ii) If only one station (station \(k\)) is automated and the automated machine is the bottleneck, then the capacity of the line is the bottleneck rate \(\mu_b = 1/t_k\).

Note that these results assume unlimited buffers between stations and hence apply to lines in which there is ample space between stations, or lines with small jobs so that a large number of jobs can be stored between stations. We can make some observations about the case where buffers are finite. However, these are limited by the complexity of finite buffer systems (the throughput analysis of nonautomated lines with finite buffers is complex, as noted by Buzacott and Shanthikumar 1993; now with automated machinery the analysis becomes even more complex because it depends not only on the buffer sizes, but also on the degree of automation as well as the worker assignment policy). Nevertheless, based on what we have shown for lines with ample size buffers, we can derive the following insights for AAP lines with limited buffer sizes:

- If the capacity of a line with ample (infinite) buffers is strictly less than its bottleneck rate, then this is also true for the same line with limited buffers. Therefore, part (i) of Lemmas 1 and 2 also hold for systems with limited buffer sizes.

- For cases in which the capacity of the line can reach its bottleneck rate with ample buffers (i.e., parts (ii) and (iii) of Lemma 1 and part (ii) of Lemma 2 hold), there is no guarantee that the same line with limited buffers also reaches its capacity. In fact, one can always set a buffer size so small that due to blocking or starvation, the capacity of the line does not reach its bottleneck rate. (Of course, this is also the case for lines without automation.) Therefore, it is of interest to inquire into when imposing a limited buffer does not degrade the line’s capacity. Here we present two such cases:

(a) If the worker is the bottleneck, both stations are automated, and both Conditions E1 and E2 hold, then the capacity of the line is the bottleneck rate \(\mu_b = 1/t_b\), even with finite buffers. The reason is that when both Conditions E1 and E2 hold, the capacity can reach the bottleneck rate under a cyclic policy, which can operate unhindered with a buffer size of one at each station.

(b) Consider the case when only Station 1 (Station 2) is automated, the automated machine is the bottleneck, and there is a finite buffer between the two stations. Letting \(X_{\mu_2}^{(m)} (X_{1}^{(m)})\) denote the random variables representing the total operation time on manual Station 2 (Station 1), we define:

**Condition E3.** \(\Pr\{X_{\mu_2}^{(m)} > X_{\mu_1}\} = 0.\)

**Condition E4.** \(\Pr\{X_{\mu_2}^{(m)} < X_{1}^{(m)}\} = 0.\)

It is easy to show that the capacity of the line is the bottleneck rate \(\mu_b = 1/t_1\) \((\mu_b = 1/t_2)\) if Condition E3 (Condition E4) holds because the line reaches its capacity under the cyclic policy.

While these results for finite buffer lines are interesting, we will focus on systems with ample buffers. We do this because: (1) systems where buffer size does not limit performance certainly exist, (2) neglecting the effects of buffers allows us to more clearly study automation issues, and most importantly, (3) buffer-sizing decisions would typically be made after automation level and position have been chosen, and so treating buffer sizes as constraints on automation decisions would rarely make sense.

Lemmas 1 and 2 suggest that in serial AAP systems there can exist a gap between the bottleneck rate and the capacity of the line. To investigate the magnitude of this gap, we developed an MDP model that obtains the capacity of a two-station line by maximizing the throughput of
the line given unlimited job availability at Station 1. The details of the MDP model are given in the online appendix. With it, we computed the gap between the capacity (maximum throughput) of the line and its bottleneck rate for several examples with different parameter settings (loading, unloading, and processing rates). Table 1 shows the gaps for one set of our test problems and clearly demonstrates that the capacity of the line can sometimes be significantly lower than its bottleneck rate.

In Case 1 of Table 1, Machine 1 is the bottleneck and there is no gap between the capacity of the line and its bottleneck rate. In Case 2, on the other hand, Machine 1 is still the bottleneck, but there is a 4.16% gap between the line capacity and its bottleneck rate. The reason is that in Case 2, the bottleneck is not as sharp as in Case 1 (i.e., line capacity and its bottleneck rate. The reason is that in still the bottleneck, but there is a 4.16% gap between the bottleneck rate. In Case 2, on the other hand, Machine 1 is there is no gap between the capacity of the line and its bottleneck rate. As Cases 3 and 5 show, this can result in a significant decrease in the capacity of the line.

In Case 1 of Table 1, Machine 1 is the bottleneck and there is no gap between the capacity of the line and its bottleneck rate. In Case 2, on the other hand, Machine 1 is still the bottleneck, but there is a 4.16% gap between the line capacity and its bottleneck rate. The reason is that in Case 2, the bottleneck is not as sharp as in Case 1 (i.e., line capacity and its bottleneck rate. The reason is that in still the bottleneck, but there is a 4.16% gap between the bottleneck rate. In Case 2, on the other hand, Machine 1 is there is no gap between the capacity of the line and its bottleneck rate. As Cases 3 and 5 show, this can result in a significant decrease in the capacity of the line.

In Case 1 of Table 1, Machine 1 is the bottleneck and there is no gap between the capacity of the line and its bottleneck rate. In Case 2, on the other hand, Machine 1 is still the bottleneck, but there is a 4.16% gap between the line capacity and its bottleneck rate. The reason is that in Case 2, the bottleneck is not as sharp as in Case 1 (i.e., line capacity and its bottleneck rate. The reason is that in still the bottleneck, but there is a 4.16% gap between the bottleneck rate. In Case 2, on the other hand, Machine 1 is there is no gap between the capacity of the line and its bottleneck rate. As Cases 3 and 5 show, this can result in a significant decrease in the capacity of the line.

### Table 1. Examples of the gap between capacity and bottleneck rate of a two-station AAP line.

<table>
<thead>
<tr>
<th>Case</th>
<th>$1/l_1$</th>
<th>$1/\mu_1$</th>
<th>$1/t_1$</th>
<th>$1/l_2$</th>
<th>$1/\mu_2$</th>
<th>$1/t_2$</th>
<th>Bottleneck</th>
<th>Bottleneck rate ($\mu_b$)</th>
<th>Capacity ($C$)</th>
<th>Gap ($\mu_b - C$)/$\mu_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.60</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>Machine 1</td>
<td>1.25</td>
<td>1.25</td>
<td>No gap</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.45</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>Machine 1</td>
<td>1.54</td>
<td>1.47</td>
<td>4.16%</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>0.40</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>Machine 1 and 2</td>
<td>1.67</td>
<td>1.52</td>
<td>8.52%</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>0.40</td>
<td>0.3</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td>Worker</td>
<td>0.83</td>
<td>0.75</td>
<td>9.72%</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
<td>0.40</td>
<td>0.2</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>Worker/Machine 1 and 2</td>
<td>1.25</td>
<td>1.02</td>
<td>18.20%</td>
</tr>
</tbody>
</table>

In this section, we assume that Station 1 is the automated station, while Station 2 is manual. We assume that jobs arrive randomly at Station 1 according to a Poisson process and that all loading and processing times are exponential. At each arrival or task completion time, the worker must decide where to work. Note that decisions are required only at these “event” times due to the memoryless property of the exponential.

#### 4.1. Automation at Station 1

In this section, we assume that Station 1 is the automated station, while Station 2 is manual. We assume that jobs arrive randomly at Station 1 according to a Poisson process and that all loading and processing times are exponential. At each arrival or task completion time, the worker must decide where to work. Note that decisions are required only at these “event” times due to the memoryless property of the exponential.

##### 4.1.1. MDP Formulation

The above assumptions allow us to formulate the problem of finding the policy that minimizes average WIP as an MDP. To do this, we define:

#### System state: $(n_1, n_2, s)$, where $n_1$ and $n_2$ are the WIP levels (including jobs in process) at the first and the second stations, respectively, while $s$ refers to the status of the automatic machine at Station 1. $s = 1$ implies that the automatic machine is performing automatic processing and $s = 0$ implies that it is not processing a job.

#### Decision epochs: consist of arrival epochs of jobs at Station 1, machine-loading completion epochs at Station 1, machine-processing completion epochs at Station 1, and job-processing completion epochs at Station 2.

#### Action space: includes (i) idling, (ii) processing a job at Station 2 (if there is a job at Station 2), and (iii) loading the automatic machine at Station 1 (if the machine is idle and the station is nonempty).

We assume that loading the automatic machine requires an exponential amount of time with mean $1/l_1$. Although in practice automated process times are usually close to deterministic for a given job type, occasional interruptions such as failures, adjustments, cleanings, and material outages will sometimes prolong the effective process time. To approximate this behavior (and keep our model tractable), we represent the automatic processing time as
exponential with mean $1/\mu_i$. We also model the manual process times at Station 2 as exponential with mean $t_i = 1/\mu_i$. Assuming that the worker can preempt a task to switch between stations (e.g., when an arrival occurs) and letting $V(n_1, n_2, s)$ be the relative value function of being in state $(n_1, n_2, s)$, the optimality equation for the MDP with the objective of minimizing the average WIP per unit time can be expressed as

$$
\frac{g}{\lambda} + V(n_1, n_2, 0) = \frac{n_1 + n_2}{\lambda} + \frac{\lambda}{\lambda} V(n_1 + 1, n_2, 0) + \frac{\mu_1}{\lambda} V(n_1, n_2, 0) + \frac{1}{\lambda} \min \left\{ \begin{array}{ll}
(l_1 + \mu_2)V(n_1, n_2, 0); & \text{idling,} \\
J_1 V(n_1, n_2, 1) + \mu_2 V(n_1, n_2, 0); & \text{loading at Station 1,} \\
\mu_2 V(n_1, n_2 - 1, 0) + \mu V(n_1, n_2, 0); & \text{processing at Station 2,}
\end{array} \right.
$$

(2)

$$
\frac{g}{\lambda} + V(n_1, n_2, 1) = \frac{n_1 + n_2}{\lambda} + \frac{\lambda}{\lambda} V(n_1 + 1, n_2, 1) + \frac{\mu_1}{\lambda} V(n_1 - 1, n_2 + 1, 0) + \frac{1}{\lambda} \min \left\{ \begin{array}{ll}
V(n_1, n_2, 1); & \text{idling,} \\
V(n_1, n_2 - 1, 1); & \text{processing at Station 2,}
\end{array} \right.
$$

(3)

where $\lambda = \lambda + l_1 + \mu_1 + \mu_2$ and $g$ is the average cost per unit time (i.e., average WIP in the system).

We note here that the exponential assumption regarding the distribution of loading and processing times is what allows us to formulate this MDP and derive an optimal policy. However, after we characterize this policy and use it to gain insights into simple lines, we relax the exponential assumption in §5 and consider more general serial lines operating under heuristic policies.

4.1.2. Structure of the Optimal Policy. In this section, we characterize the structure of the optimal policy. Omitted proofs can be found in the online appendix. We begin by justifying the MDP solution.

Theorem 1. If $\lambda < \mu_0$, then there exists an average-cost optimal stationary policy for the MDP defined by (2) and (3) that has a constant average cost. Moreover, the corresponding value iteration algorithm converges.

To characterize the structure of the optimal policy, we first require the following technical result.

Proposition 1. The value function in optimality Equations (2) and (3) has the following properties:

V1. $V(n_1, n_2, 1)$ is nondecreasing in $n_2$ for $n_1 \geq 1$ and $n_2 \geq 1$.

V2. $V(n_1, n_2, s)$ is nonincreasing in $s$ for $n_1 \geq 1$.

V3. $V(n_1, n_2, 0)$ is nondecreasing in $n_2$ for $n_2 \geq 1$.

C1. $V(n_1 - 1, n_2 + 1, 0) \leq V(n_1, n_2, 1)$ for $n_1 \geq 1$.

Theorem 2. The optimal policy for the two-station line with Station 1 automated is nonidling.

Proof. The proof follows directly from Conditions V1, V2, and V3. For example, under Condition V2, we have $V(n_1, n_2, 0) \geq V(n_1, n_2, 1)$. Multiplying both sides by $l_1$, and also adding $\mu_2 V(n_1, n_2, 0)$ to both sides, we obtain

$$
l_1 V(n_1, n_2, 0) + \mu_2 V(n_1, n_2, 0) \geq l_1 V(n_1, n_2, 1) + \mu_2 V(n_1, n_2, 0).
$$

(4)

Considering optimality Equation (2), inequality (4) implies that whenever the automatic machine at Station 1 is not processing a job ($s = 0$), then loading the machine is always a better action than idling. In a similar fashion, we can show that Condition V3 guarantees that when the automatic machine at Station 1 is not processing a job, processing a job at Station 2 is preferred to idling. Finally, we can show that Condition V1 implies that when the automatic machine is processing a job at Station 1, processing a job at Station 2 is a better action than idling. Hence, the idling action is never optimal.

To further characterize the structure of the optimal policy, we need the additional technical results of the following proposition.

Proposition 2. The value function of optimality Equations (2) and (3) has the following properties:

W1. $\mu_2 [V(n_1, n_2 - 1, 0) - V(n_1, n_2, 0)] + l_1 [V(n_1, n_2, 0) - V(n_1, n_2, 1)]$ is nonincreasing in $n_2$.

W2. $\mu_2 [V(n_1, n_2 - 1, 0) - V(n_1, n_2, 0)] + l_1 [V(n_1, n_2, 0) - V(n_1, n_2, 1)]$ is nondecreasing in $n_2$.

D1. $V(n_1, n_2, 1) - V(n_1 - 1, n_2, 1) = 0$ is nondecreasing in $n_2$.

D2. $V(n_1, n_2, 1) - V(n_1 - 1, n_2 + 1, 0)$ is nondecreasing in $n_2$.

M1. $V(n_1, n_2, 0) - V(n_1, n_2, 1)$ is nondecreasing in $n_2$.

M2. $V(n_1, n_2, 0) - V(n_1, n_2 - 1, 0)$ is nondecreasing in $n_2$.

M3. $V(n_1, n_2, 1) - V(n_1, n_2 - 1, 1)$ is nondecreasing in $n_2$.

M4. $l_1 [V(n_1, 1, 0) - V(n_1, 1, 1) - V(n_1, 0, 0) - V(n_1, 0, 1)] + \mu_2 [V(n_1, 0, 0) - V(n_1, 1, 0)] \leq 0$.

We now present the main result of this section.

Theorem 3. When the automated station is not processing ($s = 0$), then the optimal policy is a threshold-type policy with the following monotonicity properties:

• If in state $(n_1, n_2, 0)$ it is optimal to process a job at Station 2, then it is also optimal to process a job at Station 2 when the system is in state $(n_1, n_2 + 1, 0)$.

• If in state $(n_1, n_2, 0)$ it is optimal to load the automatic machine at Station 1, then loading the automatic machine is also optimal when the system is in state $(n_1 + 1, n_2, 0)$. 
**Figure 1.** Typical example of the optimal policy when Station 1 is automated.

**Proof.** Because idling is not optimal, Conditions W1 and W2 guarantee the monotonicity properties in Theorem 3. Because the details are straightforward, we omit them. □

Figure 1 illustrates the monotonic threshold policy described by Theorem 3 for the case where Station 1 is automated. The left side of Figure 1 shows the optimal policy for the case where Station 1 is not processing a job \((s = 0)\) and the right side of Figure 1 shows the case where Station 1 is processing a job \((s = 1)\).

The results of Van Oyen et al. (2001) imply that in a serial line with a single cross-trained worker and no automation, the optimal policy is to work as far downstream as possible. They label this policy the *pick-and-run policy* because under it a worker will pick up a job and run it completely through the line before returning to the front of the line for another job. The example in Figure 1 shows that automating Station 1 causes the pick-and-run policy to no longer be optimal. Because Station 1 is automated, the worker sometimes loads it up before working at Station 2. Monotonicity implies that either more WIP at Station 1 or less WIP at Station 2 makes it more attractive to work at Station 1.

### 4.2. Automation at Station 2

We now consider the case where the automatic machine is placed at the second station. Analogous to our assumptions for the previous model, we assume that the loading time of the machine at the second station is exponential with mean \(1/l_2\), and the automatic processing time is exponential with mean \(1/\mu_2\). The manual processing time at Station 1 is exponential with mean \(1/\mu_1\).

#### 4.2.1. MDP Formulation.

To formulate our model, we define:

- **System state**: \((n_1, n_2, s)\), where \(n_1\) and \(n_2\) are the WIP levels (including jobs in process) at the first and the second stations, respectively, while \(s\) refers to the status of the automatic machine at Station 2. \(s = 1\) implies that the automatic machine is performing automatic processing and \(s = 0\) implies that it is not processing a job.

- **Decision epochs**: consist of arrival epochs of jobs at Station 1, machine-loading completion epochs at Station 2, machine-processing completion epochs at Station 2, and job-processing completion epochs at Station 1.

- **Action space**: includes (i) idling, (ii) processing a job at Station 1 (if there is a job at Station 1), and (iii) loading the automatic machine at Station 2 (if the machine is idle and Station 2 is nonempty).

Again, assuming that the work is preemptable, the optimality equation for the MDP with the objective of minimizing the average WIP per unit time can be expressed as

\[
\frac{g}{\Lambda} + V(n_1, n_2, 0) = \frac{n_1 + n_2}{\Lambda} + \frac{\lambda}{\Lambda} V(n_1 + 1, n_2, 0) + \frac{\mu_2}{\Lambda} V(n_1, n_2, 0) \\
+ \frac{1}{\Lambda} \min \begin{cases} 
(m_1 + l_2)V(n_1, n_2, 0); & \text{idling}, \\
\mu_1 V(n_1 - 1, n_2 + 1, 0) + l_2 V(n_1, n_2, 0); & \text{processing at Station 1,} \\
\mu_1 V(n_1, n_2, 0) + l_2 V(n_1, n_2, 1); & \text{loading Station 2,} 
\end{cases} 
\tag{5}
\]

\[
\frac{g}{\Lambda} + V(n_1, n_2, 1) = \frac{n_1 + n_2}{\Lambda} + \frac{\lambda}{\Lambda} V(n_1 + 1, n_2, 1) \\
+ \frac{l_2}{\Lambda} V(n_1, n_2, 1) + \frac{\mu_2}{\Lambda} V(n_1, n_2 - 1, 0) \\
+ \frac{\mu_1}{\Lambda} \min \begin{cases} 
V(n_1, n_2, 1); & \text{idling}, \\
V(n_1 - 1, n_2 + 1, 1); & \text{processing at Station 1,} 
\end{cases} 
\tag{6}
\]

where \(\Lambda = \lambda + \mu_1 + l_2 + \mu_2\).
4.2.2. Structure of the Optimal Policy. We can characterize the worker’s optimal policy when the second station is automated in a manner similar to that of the previous section.

**Theorem 4.** If $\lambda < \mu$, then there exists an average-cost optimal stationary policy for the MDP defined by (5) and (6), which has a constant average cost. Moreover, the corresponding value iteration algorithm converges.

The proof of Theorem 4 is similar to that of Theorem 1 and is therefore omitted.

**Proposition 3.** The value functions in optimality Equations (5) and (6) have the following properties:

\begin{enumerate}
\item $V(n_1, n_2 - 1, 0) \leq V(n_1, n_2, 1)$ for $n_1 \geq 0$, $n_2 \geq 1$.
\item $V(n_1, n_2, s)$ is nonincreasing in $s$ for $n_1 \geq 0$ and $n_2 \geq 1$.
\item $V(n_1 - 1, n_2 + 1, 0) \leq V(n_1, n_2, 0)$ for $n_1 \geq 1$, $n_2 \geq 0$.
\item $V(n_1 - 1, n_2 + 1, 1) \leq V(n_1, n_2, 1)$ for $n_1 \geq 1$, $n_2 \geq 1$.
\item $l_2 V(n_1, n_2, 1) + \mu_1 V(n_1, n_2, 0) \leq l_1 V(n_1, n_2, 0) + \mu_1 V(n_1 - 1, n_2 + 1, 1)$ for $n_1 \geq 1$, $n_2 \geq 1$.
\end{enumerate}

We can now present the main results of this section.

**Theorem 5.** The optimal policy for the two-station line with Station 2 automated is nonidling.

**Theorem 6.** When the automated machine in nonempty Station 2 is not processing a job, the optimal policy is to always load that machine.

The proofs of Theorems 5 and 6 are very similar to those for Theorems 2 and 3 and are therefore omitted. Theorems 5 and 6 show that the optimal policy always gives priority to the automatic machine at Station 2 regardless of the amount of WIP at Stations 1 and 2. We refer to a policy that only assigns a worker to a station when there is no work at a higher-priority station as a fixed-priority policy. Note that when Station 1 is automated, the optimal policy is not a fixed-priority policy because optimal actions depend on WIP levels.

4.3. Numerical Results

To this point, we have characterized the optimal operating policy for a given automation configuration. The next question is, which station should we automate? We can (and do) use our MDP models to answer this question for simple two-station lines. However, because the optimal policy is complex, it is unlikely to find them in use in practice. Simple policies like the fixed-priority policy and cyclic policy would be more practical. Therefore, we also consider the question of where to put automation under the assumption that a fixed-priority or a cyclic policy will be used to allocate the worker to machines. We define:

\begin{itemize}
\item $WIP_{a2}$: The average WIP under the optimal policy when the second station is automated.
\item $WIP_{f1}$: The average WIP under the best fixed-priority policy when the first station is automated.
\item $WIP_{c1}$: The average WIP under the best cyclic policy when the first station is automated.
\item $WIP_{c2}$: The average WIP under the best cyclic policy when the second station is automated.
\end{itemize}

To calculate these values, we must approximate the infinite state space with a finite one. We do this by truncating the WIP levels at 60 at each station. We found that for our set of cases, truncation at a WIP level of 60 has almost no effect on the optimal average WIP. We stopped the value iteration algorithm when the error bound reached a value less than 0.001. Define

\begin{align*}
\delta_1 &= \frac{WIP_{o2} - WIP_{a2}}{WIP_{a2}}, & \delta_2 &= \frac{WIP_{f1} - WIP_{o2}}{WIP_{a2}}, \\
\delta_3 &= \frac{WIP_{c1} - WIP_{a2}}{WIP_{a2}}, & \delta_4 &= \frac{WIP_{c1} - WIP_{o2}}{WIP_{a2}}, \\
\delta_5 &= \frac{WIP_{c2} - WIP_{a2}}{WIP_{a2}}, & \delta_6 &= \frac{WIP_{c2} - WIP_{o2}}{WIP_{a2}}.
\end{align*}

Note that the optimal policy with Station 2 automated is a fixed-priority policy, so $WIP_{a2} = WIP_{f2}$, and

\begin{itemize}
\item $\delta_1$ represents the percent by which WIP increases if we automate Station 1 instead of Station 2, assuming that we use the optimal policy to allocate the worker in both cases.
\item $\delta_2$ represents the percent increase in WIP that results from using a fixed-priority policy instead of the optimal policy when Station 1 is automated.
\item $\delta_3$ represents the percent increase in WIP if we automate Station 1 instead of Station 2, but use a fixed-priority policy.
\item $\delta_4$ represents the percent increase in WIP that results from using a cyclic policy instead of the optimal policy when Station 1 is automated.
\item $\delta_5$ represents the percent increase in WIP that results from using a cyclic policy instead of the optimal policy when Station 2 is automated.
\item $\delta_6$ represents the percent increase in WIP if we automate Station 1 instead of Station 2, but use a cyclic policy.
\end{itemize}

We first investigate balanced lines. Without loss of generality, we pick a balanced line with total service times at each station equal to 1. The utilization of the worker is defined as job arrival rate times the total average manual operation time required on each job. As we mentioned before, worker utilization gives an indication of how busy the worker is with loading and unloading operations in the line. We increase the automatic processing time at the automated station from 0.1 to 0.9 in increments of 0.1, while keeping the total operation time at the station equal to 1. In other words, we vary the percent automation (i.e., $\Omega_i$, see §3) on the automated station from 10% to 90%, but adjust the arrival rate so that the utilization of the worker is
fixed at a specific level (variously chosen to be 60%, 70%, 80%, and 90%). The results under a fixed-priority policy are presented in Figure 2 and those for a cyclic policy are presented in Figure 3.

We then look at unbalanced lines with a single bottleneck with $t_b = 2$ placed at the first station and then at the second station. Because the figures for the unbalanced lines are very similar to those for balanced lines, we omit them to save space. We summarize the insights provided by this analysis in the following observations.

**Fixed-Priority Policies**

**Observation 1.** Downstream automation is more effective than upstream automation. This is true whether the optimal policy or the fixed-priority heuristic is used because $\delta_1$ and $\delta_3$ are both always positive.

When the optimal policy is implemented, the difference between downstream and upstream automation is modest (no more than 15%) for balanced lines or lines with the bottleneck at Station 2, but for lines with the bottleneck at Station 1, the difference is larger (up to 30%). In addition, the difference is relatively insensitive to worker utilization and automation level. However, when worker utilization is high and the fixed-priority heuristic is used, the difference between automating Stations 1 and 2 can be large (up to 65%).

**Observation 2.** The fixed-priority policy is more effective for lines with downstream automation than lines with upstream automation. This follows directly from Theorem 4, which implies that the fixed-priority policy is optimal when Station 2 is automated.

**Observation 3.** The error from using the fixed-priority heuristic when the first station is automated, measured by $\delta_2$ in Figure 2, is increasing in worker utilization. When worker utilization is high (90%), this error can be significant (up to 55%). Clearly, the busier the worker, the more important the allocation policy. However, when the

**Figure 2.** The effect of automation on the two-station balanced line ($t_1 = 1, t_2 = 1$)—fixed-priority vs. optimal.

**Figure 3.** The effect of automation on the two-station balanced line ($t_1 = 1, t_2 = 1$)—cyclic vs. optimal.
automation level is high (above 60%), which is common in practice, the error resulting from the fixed-priority heuristic is small (less than 10%).

Observation 4. When the first station is automated, the automation level at which the maximum error from using a fixed-priority policy occurs is nonincreasing in worker utilization. The reason is that as the worker gets busier, loading the automated first station becomes more important. Hence, the automation level at which the fixed-priority policy switches from prioritizing Station 2 to prioritizing Station 1 (and hence the point of maximum error) decreases with worker utilization.

Cyclic Policy

Observation 5. When a cyclic policy is implemented, downstream automation is also more effective than upstream automation. As Figure 3 shows, \( \delta_6 \) is always nonnegative. The difference decreases in percent automation and worker utilization. In fact, as we noted, when worker utilization is high, there is little difference between automating Stations 1 and 2 if percent automation is larger than 50%.

Observation 6. The best cyclic policy performs close to the optimal policy when the system has low-percent automation and the worker is not heavily utilized. For example, as Figure 3 shows, \( \delta_4 \) and \( \delta_5 \) are no larger than 10% in systems with worker utilization of 70%, and percent automation less than 40%. However, the performance of the cyclic policy deviates quickly from the optimal policy as percent automation and worker utilization increase. When worker utilization reaches 90%, even with 40% percent automation the relative performance between the cyclic policy and the optimal policy is larger than 160%!

5. Three-Station Lines

The above two-station lines lead to tractable models and clean insights. However, most real-world systems involve more than two stations. So, to determine the extent to which our observations carry over to larger systems, we now consider three-station serial lines with one fully cross-trained worker. Because an MDP model of such lines is too cumbersome to solve and the resulting policy too complex for practice, we restrict our attention to easily implementable heuristics (i.e., fixed-priority policies and cyclic policies) and use simulation to evaluate their performance.

As in our two-station lines, we assume that arrivals to the three-station line follow a Poisson process, machines never break down, and there is ample buffer space between stations. Job operation times on machines may consist of three parts: manual loading, automatic processing, and manual unloading times. Operators are required during manual operations, but not automatic ones. For realism, we assume that loading and unloading operation times are stochastic, but automatic operation times are deterministic. Our numerical results in the rest of the paper are based on simulation of systems in which loading and unloading operation times follow Erlang-4 or Erlang-1 (i.e., exponential) distributions. Our simulation model is developed for general lines in which any number of stations may be automated. However, we model manual stations by setting the automatic processing time of that station to zero. This creates a manual operation time that is the sum of two Erlang random variables. Because this had the effect of reducing the variability of manual process times, we repeated our simulation for cases where the manual operations were the sum of two exponential random variables, but found that this did not affect our observations.

Without loss of generality, we pick a balanced line with total operation time at each station equal to 10. For convenience, we denote this line by (10, 10, 10). We also consider unbalanced lines with a bottleneck requiring total job time of 20 placed at various positions in the line. These are denoted by (20, 10, 10), (10, 20, 10), and (10, 10, 20).

5.1. Impact of Automation

To examine the effect of automation level on the choice of operating policy, we start with the balanced line (10, 10, 10) and increase the amount of automation (i.e., the automatic processing time) on all stations from 1 to 9, while keeping total operation times at each station, and the arrival rate, constant. For each automation level, we compare the performance (average WIP) of all fixed-priority policies and cyclic policies. We do the same for the unbalanced lines.

To compare policies, we ran a simulation model for each case with 20 replications containing about 25,000 arrivals. Furthermore, we made use of a warm-up period to avoid the effect of initial bias, and used the common random number (CRN) technique across different policies and automation levels. Different random number streams were used for loading and unloading times at different stations to ensure independence. Because the results for balanced and unbalanced lines were similar, we only present the balanced-line results. Figure 4 shows the results for one of several cases that we studied; for this example, the job arrival rate is \( \lambda = 0.028 \).

Observation 7. The diminishing-return law holds with respect to WIP reduction from additional automation under both the fixed-priority and cyclic policies. In particular, when a machine instead of the worker becomes the bottleneck, further automation has little effect on WIP. Figure 4 shows that when the amount of automation reaches 6.67, so that the total amount of manual work is less than 10, the WIP curves become very flat. This is because when the worker is no longer the bottleneck, there is little queuing for additional automation to reduce. Note that as job arrival rate increases, WIP also increases. For example, for arrival rates larger than \( \lambda = 0.028 \), WIP at automation level 90% (i.e., nine minutes of automation) would be larger than 1.
Observation 8. The performance of the two cyclic policies 1-2-3 and 1-3-2 in three-station lines is very similar, while that of a fixed-priority policy is predominantly determined by which station is given the least priority. The reason behind this is that when a station is given the last priority, most of the total WIP in the system will end up at that station. Therefore, the first and second-priority stations will have a significantly lower WIP. Under these circumstances, the order in which these two stations are prioritized will have little effect on the WIP level in the system.

However, when a machine is the bottleneck, rather than the worker, there is little difference between the performance of the different fixed-priority policies. The reason is that when the worker is not highly utilized, it does not matter much in what order she attends the machines because they will all be covered without significant delay.

Observation 9. The performance of the best fixed-priority policy is always better than the performance of the best cyclic policy. As Figure 4 shows, regardless of the levels of automation, the performance of the best priority policy is never worse than that of the best cyclic policy. Similar insight is confirmed in unbalanced lines. In fact, we found that in the (20, 10, 10) line where automation is presented, the best fixed-priority policies may outperform the cyclic policies with a difference as large as 14%.

5.2. Impact of Automation Position

We now return to the question of which station of a manual serial line is the best candidate for automation. Of course, we recognize that in practical settings technological or other considerations may restrict the options for selecting stations to automate. However, by assuming that all stations are considered for automation, we investigate the factors that make some automation configurations more attractive than others. This provides insights that can help decision makers prioritize automation options when choices exist. We address this issue via simulation experiments that have the same inputs and scenarios as the previous section except:

- Instead of fixing the arrival rate, we now fix worker utilization to various levels—60%, 70%, 80%, and 90%.
- We now apply varying amounts of automation to only a single station.

We consider systems under fixed-priority and cyclic policies. For the former, we suppose that the worker will follow the best fixed-priority policy for each automation configuration. Hence, for every scenario we investigate the performance of the best fixed-priority policy (among all six possibilities) and plot the average WIP resulting from the best policy for that scenario. We do the same for cases under cyclic policies to obtain the best cyclic policy.

The results for the balanced line (10, 10, 10) are given in Figures 5 and 6 for worker utilization levels of 60% and 90%, respectively. We can summarize the insights from this analysis in the following observations.

Observation 10. When the best fixed-priority policy is adopted, downstream automation is more effective than upstream automation. In contrast, when the best cyclic policy is adopted, upstream automation is more effective than downstream automation.

As we have shown in §4.3, in a two-station line where the operation on the automated machine consists of only loading and automatic processing, downstream automation is more effective than upstream automation, regardless of whether a fixed-priority or cyclic policy is implemented. This is different from what we observe here. The reason is that here the job processing time in the automated station also includes an unloading operation. When the third station is automated and a cyclic policy is implemented, the addition of an unloading operation may cause a job that has completed automatic processing at the third station to wait for a long time (i.e., for at least the sum of the total manual operation times on the first two stations) before the worker
Figure 5. The impact of position of automation—worker utilization = 60%. Left: fixed-priority policy; Right: cyclic policy.

returns to that station and releases the job out of the system. Because our objective is to minimize the average WIP in the line, automating the downstream station is not effective when a cyclic policy is used. However, this does not happen when a fixed-priority policy is implemented because the server always gives highest priority to the automated third station. In other words, when automatic processing of a job in the third station is completed, the worker immediately unloads and releases it. This serves to reduce the WIP in the system.

Observation 11. When there is only one automated machine in the line, placing automation downstream and using the best fixed-priority policy outperforms placing automation upstream and using the best cyclic policy. As Figure 5 shows, WIP in a balanced line when the third station is automated and the best fixed-priority policy is implemented never exceeds that of WIP when the first station is automated and the best cyclic policy is implemented. In fact, the difference can be as large as 12%. A similar pattern is found in Figure 6 and we also observed the same phenomenon in lines with various levels of automation and unbalance.

We also studied unbalanced lines (20, 10, 10), (10, 20, 10), and (10, 10, 20). The results are similar: Downstream automation is more effective than upstream automation, regardless of where the bottleneck station is located in the line. This implies that when making decisions concerning where to place automation, automating a bottleneck station is not necessarily more effective than automating a nonbottleneck station.

Figure 6. The impact of position of automation—worker utilization = 90%. Left: fixed-priority policy; Right: cyclic policy.
Table 2. Experiments on the effect of automation concentration in a three-station AAP line.

<table>
<thead>
<tr>
<th>Case #</th>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
<th>$\omega_3$</th>
<th>Case #</th>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
<th>$\omega_3$</th>
<th>Case #</th>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
<th>$\omega_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4.5</td>
<td>0</td>
<td>4.5</td>
<td>12</td>
<td>0</td>
<td>4.5</td>
<td>4.5</td>
<td>18</td>
<td>4.5</td>
<td>4.5</td>
<td>0</td>
</tr>
</tbody>
</table>

5.3. Impact of Automation Concentration

To investigate the impact of automation concentration in an AAP line, we considered a three-station line with one fully cross-trained worker and total operation time at each station of 10 minutes ($t_i = 10$, $i = 1, 2, 3$). In three different experiments, we distributed nine minutes of automation among the three stations. Table 2 illustrates different cases of each experiment.

**Experiment 1.** Consider cases where the automation is and is not concentrated at Station 2. In Cases 1, 2, and 3, Station 2 has the highest amount of automation, but in Cases 5 and 6, Station 2 has the least amount of automation. The left side of Figure 7 shows the WIP for each case of Experiment 1 under different worker utilizations and leads us to a similar observation under both the cyclic and fixed-priority policies.

**Observation 12.** In AAP lines where the worker is the bottleneck, concentrated automation is more effective than distributed automation.

As we can see from the left side of Figure 7, when a fixed-priority policy is adopted, Case 1, where all nine minutes of automation are concentrated at Station 2, performs better than all other cases, which represent more distributed configurations. In contrast, Case 4, in which automation is evenly distributed among stations (i.e., no concentration), has the worst performance. Furthermore, the benefit of concentrated automation is more significant when worker utilization is high. As the left side of Figure 7 shows, when the worker is 60% utilized, there is little difference between all configurations; but when the worker is 90% utilized, the difference between the most distributed and the most concentrated cases can be 170%.

The right side of Figure 7 illustrates the same behavior, and shows that when a cyclic policy is adopted, concentrated automation is also more effective than distributed automation, although the difference is less pronounced than when a fixed-priority policy is used.

The main reason that concentrated automation is more effective than distributed automation is because the efficiency of the system depends on avoiding idling by the worker (who is the bottleneck). The more machines that are automated and the more balanced their automation levels, the more likely the worker is to be forced to idle while waiting for automated machines to finish.

**Figure 7.** The impact of automation concentration—Experiment 1. Left: fixed-priority policy; Right: cyclic policy.
Experiment 2. This experiment performs a study similar to that of Experiment 1, but focuses on concentrated automation at Station 1. Figure 8 shows the WIP for each case of Experiment 2 under the fixed-priority and cyclic policies.

Observation 13. System performance is a function of both automation concentration and automation position. The left side of Figure 8 shows that when a fixed-priority policy is adopted, the performance of Case 12, where all automation is distributed to Stations 2 and 3, is as good as that of Case 7, in which all automation is concentrated at Station 1, and both outperform all other configurations. At the same time, although in Case 9 automation is more concentrated than in Case 10, Case 10 has better performance when worker utilization is high. Note that upstream Station 1 in Case 10 has a lower amount of automation than in Case 9. This confirms what we observed in §5.2, namely, that automation at downstream stations tends to be more effective than automation at upstream stations in an AAP line under the fixed-priority policy.

We observed a similar pattern with respect to upstream stations for systems under the cyclic policy. However, the impact is less pronounced.

Experiment 3. This experiment focuses on concentration of automation at Station 3. Because the results are similar to those of Experiments 1 and 2, we omit their discussion.

6. Conclusions and Further Work

Design and control of manufacturing systems with automatic equipment and cross-trained (agile) workers are significantly more difficult than design and control of systems without automation. This paper represents a first step toward understanding the characteristics of these challenging agile automated production (AAP) systems. By studying two- and three-station lines consisting of a mixture of manual and automated machines and staffed by a single cross-trained worker, we are able to make several general observations regarding the AAP systems:

1. Planning. In capacity planning involving serial AAP systems, one must consider that an AAP system may have capacity substantially lower than its bottleneck rate. Furthermore, the actual capacity can be very difficult to compute. We have shown that even for simple cases (i.e., two-station AAP lines), the gap between the capacity and the bottleneck rate can be as large as 18%.

2. Design. Some of the major factors that must be considered in the design of AAP systems are:
   - Performance of serial AAP systems (i.e., manufacturing cells) is sensitive to the level, position, and concentration of automation. We have shown that if a fixed-priority policy is implemented, then downstream automation is more effective than upstream automation. On the other hand, if a cyclic policy is implemented, then upstream automation can be more effective than downstream automation (i.e., when the last station has an unloading operation). Furthermore, with either policy, concentrated automation is more effective than evenly distributed automation.
   - When worker utilization is low, increasing the level of automation in the line does not significantly contribute to WIP (or cycle time) reduction in the line.
   - Although balanced lines may be ideal for lines with no automation, they may not be a suitable design for AAP lines. This is because in balanced lines, when the worker is not the bottleneck, all machines in the line become bottlenecks. We have shown that when more than one machine acts as a bottleneck, the gap between the capacity of the line and the bottleneck rate increases.

3. Control. Performance of an AAP line is sensitive to the worker allocation policy. We have shown threshold structures for the optimal policies in lines with two stations. However, even for those lines, the optimal policy is too complex for practice. Therefore, we restricted our attention to easily implementable fixed-priority and cyclic policies.
We have shown that if a fixed-priority policy is implemented, then there is a strong incentive for the worker to work as far downstream as possible. However, particularly when worker utilization is high, the worker should also give preference to loading automated machines before tending to manual ones. This second observation can override the first if automated machines are placed upstream. Moreover, we have shown that for balanced lines with only one automated machine, placing automation downstream and using the best fixed-priority policy outperforms placing automation upstream and using the best cyclic policy. Therefore, we conjecture that designing AAP lines so that automation is weighted toward the end of the line, and using a policy that favors downstream work, is a good overall approach.

Because this paper is an early step in the study of AAP systems, it is necessarily restricted to simple systems. To see whether our insights are indeed robust in more general systems, further research is needed into systems with multiple product types, multiple routings, multiple workers, and other real-world features, such as machine failures, yield loss, rework, batching, and partial cross training. Given the growing importance of AAP systems in industry, such research would be of great practical significance.

References


Monden, M. 1993. Toyota Production System: An Integrated Approach to Just-In-Time. Institute of Industrial Engineers, Atlanta, GA.


