Manufacturing & Service Operations Management

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Management of Worksharing Systems

John O. McClain, Kenneth L. Schultz, L. Joseph Thomas,

Management of Worksharing Systems

John O. McClain • Kenneth L. Schultz • L. Joseph Thomas
S.C. Johnson Graduate School of Management, Cornell University, Ithaca, New York 14853
Kelley School of Business, Indiana University, Bloomington, Indiana 47405
S.C. Johnson Graduate School of Management, Cornell University, Ithaca, New York 14853

Worksharing occurs in serial manufacturing when machines are not uniquely assigned to workers. For example, two workers can operate three machines by alternating usage of the middle one. In some other worksharing systems operators move down the line carrying an item (or batch) with them, working on it at each machine until they are met by another worker who is coming back upstream. The work is then handed off *even if the operation is underway.* "Bucket Brigade" and "TSS" are worksharing systems based on this idea.

This paper examines worksharing in a variety of situations, including unequal work content across machines, uncertain processing times, unequal workers, handoffs with and without preemption, and a range of machine-to-worker ratios. Some systems restrict workers to "zones" of machines. Work zones must overlap if sharing is to occur. The appropriate size of the overlap is shown to depend on the circumstances. Inventory-based rules to control access to shared machines are demonstrated to increase productivity when processing times vary. Worker sequence is found to be quite important; Slowest-to-Fastest is recommended for some situations, but is shown to perform poorly in others. Finally, the issue of whether or not to allow preemption is shown to have a large impact on productivity if inventory is not allowed, and to strongly affect the choice of designs when inventory is allowed. With and without preemption, proper combinations of worker sequence, zone size, and production control rules are shown to nearly eliminate idle time.
(*Worksharing; Serial Production Lines; Cross Training; Bucket Brigade*)

1. Introduction

Reducing idle time of a limiting resource is the key to increased productivity. When labor is the limiting resource in a serial production facility, *worksharing* is one way to reduce idle time. Cross training workers to share responsibility for some machines brings improvement in two ways. First, it gives the system flexibility to counteract process variability, which may be caused by uncertainty or by mixed-model production. For example, if variation causes the first worker to fall behind, the second worker can compensate by doing "more than her/his share" and vice versa. Second, worksharing allows the line to be balanced by alternating which worker does a particular operation, effectively splitting the shared tasks. In effect, *the line balances itself.* Bartholdi and Eisenstein (1996) describe one environment in which self-balancing occurs. How these effects are manifested in a variety of factory environments is the subject of this paper.

In the factories we are studying, workflow is serial and each machine has a unique operation requiring one worker (e.g., a sewing machine in garment manufacture). The layout is either a straight line or U-shaped, and workers carry a single piece (or a small batch) as they move from one machine to the next. Upon reaching the end of an assigned zone the worker returns to the top of her/his zone and begins again. Neighboring workers are cross-trained so that they can share responsibility for machines where the zones overlap.

Many worksharing systems use *preemption* to avoid

1523-4614/00/0201/0049$05.00
1526-5498 electronic ISSN

idle time. Preemption occurs when a worker is returning (empty-handed) toward the beginning of her/his zone and encounters a shared machine that is being used. Rather than waiting for the operation to be completed, the arriving worker takes over the task without stopping the process. Careful training is required to exchange operators in mid-process without danger to the workers or to the product.

Worksharing systems described in the literature (see §2.1) have been typified by low processing time variability and equipment that is inexpensive relative to the price of labor. As a consequence they can operate economically with low inventory and a relatively high ratio of machines to workers. As with any new system, there are many other environments in which worksharing might be implemented if research shows that it will increase efficiency. This paper examines an environment in which processing time variability challenges the flexibility of the system and low machine-to-worker ratio limits the amount of sharing that is possible. In addition, we consider situations in which workers may not preempt one another. Section 2.2 gives examples of this environment from the literature and from real factories, and describes the roles that additional work-in-process inventory (WIP) can play.

The research objective of this paper is to find designs that achieve near-maximal output in circumstances where previously-described worksharing systems do not perform well. Two environmental parameters are explored: machine-to-worker ratio and variability of processing time. This establishes a region in which worksharing systems do not perform well without additional inventory and controls. Then, holding constant those two environmental parameters, three primary design issues are explored: size of the zones (limiting the movement of workers), control rules based on WIP (directing the movement of workers), and sequence of workers.

Section 2 describes previous research related to the objective, describes the research issues, and concludes with a detailed description of the worksharing system being studied and definitions of terms. Section 3 gives the methodology, including the ranges of parameters, a test bed and a simulation program used to evaluate alternatives. Section 4 is devoted to systems with preemption. It is organized around a set of observations based on previous literature and characteristics of worksharing systems. Each observation is accompanied by a rationale based on the characteristics of the system. From this we draw conclusions about elements that cause difficulty and suggest tools the system designer may use to assure a well-functioning production line. Section 5 investigates systems without preemption, following the same format as § 4. The last section summarizes the results and suggests directions for additional research.

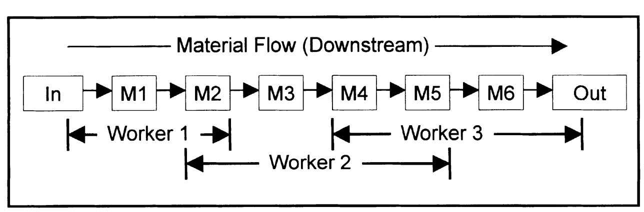
2. Research, Issues, and Definitions

A typical serial worksharing system is shown in Figure 1. There are fewer workers than machines, and each worker operates within a zone. The machines have unique operations that require one worker to be present. Some machines are idle at every point in time. Output rate is determined by how effectively the workers are utilized. In Figure 1 the work zones overlap. The first two workers share M2, and workers 2 and 3 have a two-machine overlap, sharing M4 and M5. This section describes previous research on this kind of system, lays out research issues that arise in new application environments, and concludes with definitions of the terms we will use.

2.1. Early Research and Application

TSS (Toyota Sewn-products System) is a manufacturing system based on worksharing. Operators move down the line carrying an item (or small batch) with them and working on it at each machine in an assigned zone until they are preempted by another worker who is coming back upstream. Moore (1995) referred to this as "bump-back modular" design. Bischak (1996) and Zavadlav et al. (1996) describe research and application of TSS and variations on the rules that guide the workers.

Figure 1 A Serial Production System with Worksharing Zones



"Bucket brigade," described by Bartholdi and Eisenstein (1996), differs from TSS in two ways. First, although both allow preemption, bucket brigade does not restrict workers to zones of machines. Second, when a bucket-brigade worker finishes an operation and finds the next machine in use, (s)he must stand idle and wait until the next worker finishes. TSS has no such restriction (e.g., the worker may leave and commence preemption or return toward the beginning of her/his zone). Under some conditions, Bartholdi and Eisenstein showed that the bucket-brigade system is self-balancing, by which they mean that the points where items are transferred from one worker to the next are determined by the workers' speed and tend to be stable. However, they also showed the surprising result that worker movement might be chaotic under other (quite reasonable) circumstances, even with deterministic processing times. This serves warning that care is needed in designing worksharing systems and provides motivation for this research.

2.2. Research Issues

There are many applications in which processing times vary substantially, and much literature has been devoted to studying these cases. (See Muth 1973, Knott and Sury 1987, and Conway et al. 1988, for example.) With low variability, Bartholdi and Eisenstein (1996) showed that the bucket-brigade system with Slowest-to-Fastest worker sequence always produces a repeating work cycle, so that workers tend to remain in zones of machines. With high variability there is no guarantee that these zones will emerge naturally.

The usual way to deal with uncertainty is to provide inventory buffers between operations. Bischak (1996) and Zavadlav et al. (1996) found that very small buffers are sufficient to avoid most idle time, and that there are sharply diminishing returns to additional buffer capacity. This happens with identical machines, and also when one machine has a longer processing time than the others. It mirrors the findings of Conway et al. (1988) for systems without worksharing, and Ostolaza et al. (1990) and McClain et al. (1992) for systems with stationary workers who share tasks through duplicate tooling. In each of these studies, workers were presumed to be identical, a situation very different from the applications reported by Bartholdi and Eisenstein.

However, there are situations in which small buffers cannot prevent idle time. Zavadlav et al. (1996) showed that a low worker-to-machine ratio can cause idle time even with inventory in the system. They explained that task sharing is less effective in offsetting process variation when there is substantial competition for machines. Facing a similar dilemma, Ostolaza et al. (1990) demonstrated that small buffers become much more effective when their inventories are used to indicate which worker should do a shared task, and developed a *buffer management rule* based on that concept. In this paper we explore the usefulness of a similar rule in serial worksharing lines.

Preemption is used in a variety of environments, some of which are described in Bischak (1996), Bartholdi and Eisenstein (1996), Moore (1995), and Zavadlav et al. (1996). However, in some work environments it is impractical or impossible to preempt a worker in the midst of an operation. Bischak (1996) mentions this issue. Several reasons can lead to this decision. Safety may prevent exchange of operators while the machine is running. Time lost while exchanging workers may be prohibitive. (When process times can be a minute or two, a few seconds of time loss can be too much.) The risk of incurring a defect during the exchange may be excessive. In some cases it is simply impossible to exchange workers in mid-operation.

In practice, preemption does not occur for a variety of reasons. In an automobile parts manufacturer there was no rule against preemption, but it never occurred because the tasks were very short. The time lost during preemption would have been a large fraction of the time to complete the task. In another application, a sporting goods manufacturer produced a variety of apparel and related items, including hockey pads, team shirts, and baseball gloves. They had 20 lines and used machine-to-worker ratios from 2.0 to 4.0. Lines were realigned and products were changed periodically by the engineering staff. In some cases they decided to work without preemption for certain tasks, based on quality considerations. In other cases it did not occur because some workers did not like to be preempted.

Thus, as the use of worksharing expands into new applications, researchers and practitioners need information about its performance without preemption.

Without preemption, workers must take turns performing the entire operation on shared machines, and the frequency with which a worker performs a shared task is the key to balancing the line. Maintaining the correct frequency may not be easy, even with deterministic processing times, and substantial idle time may occur. This paper examines the "NoPreempt" situation in detail, and explores the use of buffer management to control the use of shared machines. In an alternative approach, applicable when preemption is expensive, but not prohibited, Bartholdi et al. (1995) study the use of batches to reduce the frequency of preemptions. Both approaches (batches and buffer management) cause higher inventory levels and additional handling of materials.

As mentioned above, Bartholdi and Eisenstein (1996) found that if the workers are sequenced from Slowest-to-Fastest, then a Bucket-Brigade line is self-balancing. At first examination it seems that a slow worker at the front of the line would slow the entire operation by failing to provide material fast enough for the other workers. However, preemption tends to prevent this. Whenever a worker needs a new item, a chain-reaction of preemption occurs that culminates by interrupting worker 1 and sending her/him to start a new item on the first machine or even temporarily moving her/him off of the line. However, if preemption is not allowed, Slowest-to-Fastest may not be the best sequence for the workers. This paper explores the issue of worker sequence with and without preemption, under a variety of circumstances.

In summary, this paper extends the research on serial worksharing systems to environments with variation of processing time, low machine-to-worker ratios, unequal workers and unequal machines, both with and without preemption. Tools for designing the worksharing lines include buffers for work-in-process storage, using inventory to control who performs the shared tasks, zone size and worker order. Output rate of the system is the criterion. These design tools will now be explained in detail.

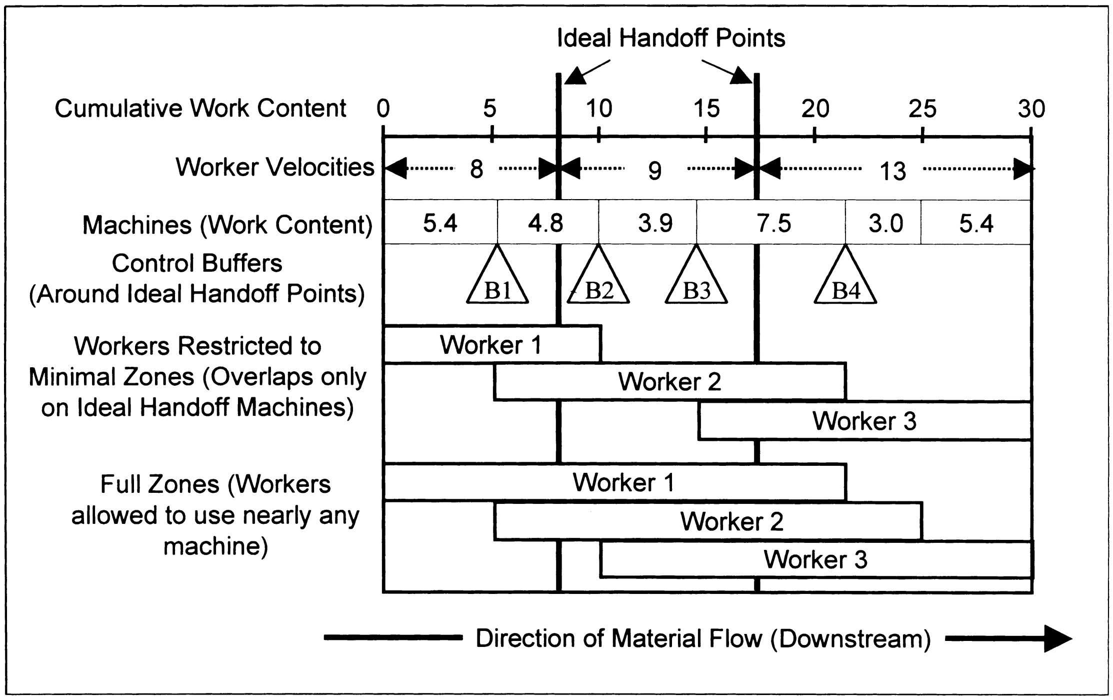
2.3. Definitions: Work Zones and Control Buffers

In this paper, control of a worksharing system involves two concepts: work zones and control buffers. Zones place absolute limits on worker movement. Control buffers govern the sharing of machines where zones overlap. Figure 2 illustrates where the control buffers are placed in the systems we are studying, and shows two alternative sets of work zones. Definitions of the terms in the figure and other terms used in this paper follow.

Definitions

- A *Zone* is the set of contiguous machines that a worker is permitted to use.
- *Shared Machines* occur within an *overlap* of neighboring zones.
- *Minimal Zones* are the smallest zones for which the line can be balanced. Each overlap contains (at most) one machine—only the *Ideal Handoff Machines* (defined below) are shared.
- *Full Zones* allow workers to use (almost) any machine. With W workers and M machines, the Full Zone for worker i includes machines i through $M - W + i$.
- *Maximum (Max) Zones* allow downstream workers to preempt even if doing so pushes a worker off of the line. This is used in the "Bucket Brigade" system. Note that not all machines are available to all workers, since only upstream preemption is allowed in the systems studied here; the last machine is available only to the last worker, for example.
- *Control Buffers* are inventory storage buffers (between machines) for which a *Control Level c* governs the movement of workers by the following *Control Rule:* If inventory is c or less, work upstream of that buffer; otherwise, work downstream. The rule is invoked when a worker is about to pass the control buffer in either direction. However, the workers are trained to disregard the control rule to avoid preempting another worker and (most importantly) to avoid idle time.
- *Uncontrolled Buffers* are the same (technically) as having control level $c = 0$. That is, whenever there is inventory in any amount, work downstream of the buffer. Uncontrolled buffers accumulate inventory when the next machine is busy or is not in the worker's zone.
- *Upstream and Downstream Controls:* In Figure 2, worker 2 may have controls at buffers B2 and B3. B2 is the upstream control buffer for worker 2 since it controls entry into the upstream neighbor's minimal zone.

Figure 2 **Illustration of Zones and Control Buffers**



Similarly, B3 is worker 2's downstream control buffer. The first and last workers have at most one control.

• *Buffer Capacity* is an absolute upper limit on the inventory in that location. This limit may not be violated under any circumstances.

• *Velocity of Worker i, v_i,* measures the worker's capability. If there is no idle time, each worker's output is proportional to his or her velocity.

• *Work Content of Machine i, τ_i,* is the average processing time on a machine by a worker with velocity 1.0. *The total work content* for a line of *M* machines is:

$$T = \sum_{i=1}^{M} \tau_i. \tag{1}$$

• *Preemption* occurs when an item is relinquished to the downstream neighbor *during an operation;* the downstream worker is said to have preempted the upstream worker.

• *Ideal Handoff Points* are the points in the process where preemption must occur to avoid idle time without inventory, if processing times were constant. Measured on a continuous time scale, the position of an Ideal Handoff Point is usually *within* an operation. In Figure 2, the distance h_i between Ideal Handoff Points is proportional to the worker's velocity:

$$h_i = Tv_i \Big/ \sum_{j=1}^{W} v_j, \tag{2}$$

in which *T* is the total work content per item and there are *W* workers.

• *Ideal Handoff Machines* are those containing Ideal Handoff Points. To balance the line, the Ideal Handoff Machines must be shared.

• *Efficiency* is the actual output rate divided by the maximum possible:

$$Efficiency = (Output\ Rate)\ T \Big/ \sum_{j=1}^{W} v_j. \tag{3}$$

Figure 2 illustrates both Minimal and Full Zones. In both cases workers 1 and 2 share the second machine. If there were no variation in processing times and no idle time, worker 2 would take over at the first Ideal Handoff Point, after slightly more than half of machine

2's operation was complete. If there is variation in processing time, worker 2 may not be ready to take over at the Ideal Handoff Point so worker 1 would continue processing on machine 2. With Minimal Zones worker 1 would go no farther. With Full Zones worker 1 would be allowed to continue processing on machines 3 and 4 (but not 5 or 6), if necessary.

The form of the control rules is consistent with the concept of sharing operations. Inventory in a control buffer is analogous to a speedometer. High inventory suggests that the downstream worker has fallen behind, and the rule encourages extra effort downstream to remedy the situation. Low inventory signals the opposite. Both of the resulting actions may be interpreted as helping a worker who has fallen behind.

Using these definitions, in §§4 and 5 we present "Observations" that are supported by theory (including literature), common sense, and/or simulation results. These are not proven mathematically because the systems are too complex. We will present our arguments and simulation results so readers can evaluate the evidence themselves. In our opinion, the most valuable "Observations" are those that are at least partially counterintuitive or that show limitations of previous research in the situations studied here. First, however, we describe the methodology including our parameter choices, a test bed used for comparisons, and the simulation program used to evaluate system performance.

3. Methodology

Our plan of attack is to investigate worksharing in realistic environments in which worksharing is feasible but the bucket brigade system does not operate well. The reason for this plan is that we have seen situations of this sort, and expect that many more will surface as the benefits of worksharing become known. We will show in §4 that low machine-to-worker ratios and moderate-to-high processing time variation characterize that environment. This section presents ranges for each of the design parameters together with rationales for those choices, followed by a description of the test bed and the simulation.

3.1. Ranges of Parameters

Selection of parameter ranges is a crucial part of our research design. There are too many possible issues (buffer size, control methods, worker order, amount of task variation, worker speeds, work content per machine, worker to machine ratio, etc.) to allow a full-factorial design. Below, we describe two different reasons for parameter choices: (1) The values are realistic, based on literature or our personal observation, and (2) we want to study particular ranges to see the limits of worksharing. Some parameters are held constant for this paper, some are varied one- or two-at-a-time, and others are explored using a full-factorial design. We always attempt to give a clear justification, so readers can interpret the results for themselves.

Parameters Held Constant

• *Three Workers.* This is the smallest system for which there is a worker who has both upstream and downstream neighbors. Having two neighbors could be an important complicating factor.

• *Location of Buffers on Both Sides of the Ideal Handoff Machines.* To prevent interference between workers, the most effective positions for inventory (and inventory-based controls, if any) should be at the ideal handoff points, which is where workers expect to meet.

• *All Downstream Control Buffers have the Same Control Level.* It seems reasonable to have the same rule for all workers, and this reduces the number of options to explore.

• *All Upstream Control Buffers have the Same Control Level.* Same rationale. (The control level may differ between upstream and downstream control buffers.)

• *All Control Buffers have the Same Storage Capacity.* Conway et al. (1988) showed that the best buffer capacity is nearly independent of position in the line, and that the penalty for using equal buffers is very slight.

• *Uniform Processing Time Distribution.* Literature has shown that, holding constant the coefficient of variation, the shape of the processing time distribution makes little difference in serial lines. (See Muth 1973, and Hendricks and McClain 1993.) Also, the uniform distribution's CV can be chosen between 0 and 0.577, which is sufficient for our purposes. The limits of the uniform distribution are $\mu(1 \pm CV\sqrt{3})$.

Parameters Varied Using Incomplete Designs

• *Processing Time Coefficient of Variation 0.0 to 0.5.* Bartholdi and Eisenstein (1996) report CV = 0.1 for one worker. Muth (1973) suggests 0.1 or lower for an organized service. Knott and Sury (1987) report values ranging from 0.22 to 0.57 for light assembly tasks.

• *Machine-to-Worker Ratio 2.0 to 6.0.* Moore (1995), Proper and Plattus (1993), Berg et al. (1996), and Bartholdi and Eisenstein (1996) report ratios between 2.0 and 5.5. Ratios between 2.0 and 4.0 were used in a factory we studied.

• *Storage Capacity of the Control Buffers: 0, 2 or 4.* Zero is used in some real systems (Moore 1995). Research on in-process inventory concludes that very small buffers are sufficient (Conway et al. 1988). Section 4.3 demonstrates that this range is sufficient for this study.

• *Worksharing Rules:* What to do when a worker overtakes (that is, catches up to) another (wait versus leave), and use of Maximum versus Full Zones.

Parameters Varied in a Full-Factorial Design

• *Two Worker Sequences: Slowest-to-Fastest and Fastest-to-Slowest.* These are two extreme cases. Nice properties for the former are cited in §2 for a system with preemption.

• *Two Zone Sizes: Full or Minimal.* These are two extreme cases among the ones we have encountered in practice.

• *Two Choices of Control Levels: None or Half-Full.* Bucket brigade systems operate without controls. Half-full controls have worked well in similar systems (Ostolaza et al. 1990).

• *Two Preemption Possibilities: Required or Not Allowed.* Both are discussed in §2.

• *Five Worker-Velocity Profiles.* Chosen to provide two extreme cases, the profiles include a case in which one worker is 3 times as fast as another, a case with equal velocities, and three cases between. This range is similar to that reported in practice by Bartholdi and Eisenstein (1996) who give velocities ranging from 0.5 to 1.6 with most between 0.8 to 1.3.

• *Twenty-Five Serial Lines of "Unequal" Machines.* Unequal machines (with equal processing rates but different work content) can be difficult to manage. For example, a zone with several "small" machines gives workers more flexibility than one with few "large" machines. The details are given in §3.2.

3.2. The Test Bed of Factories

Analyses of cases with and without preemption are described in §§4 and 5, respectively. Each section begins with a limited design (identical machines, one unequal-worker-velocity profile) to explore ideas that are strongly suggested by the literature or by common sense. Then the scope is limited to lines of 6 machines with CV = 0.3 (rationale given in §4.2) and expanded to a test bed of 125 cases with unequal machines and workers. The average work content is 5.0 per machine, and there is at least one machine with mean work content that is 50% shorter or longer than the average machine. This was done to assure that all of the lines had comparable ranges of machines. The following algorithm used M uniformly distributed random numbers $\{r_i\}$ to generate the mean work contents $\{\tau;_i\}$ given in Table 1:

$$\bar{r} = \text{Average}\{r_i\},\ R = \text{Max}\ \{r_i\},\ r = \text{Min}\{r_i\},$$

$$a = \max\ \{R - \bar{r}, \bar{r} - r\},\ \tau_i = (2.5/a)(r_i - \bar{r}) + 5. \quad (4)$$

Despite use of uniform variates, the distribution of

Table 1 **Parameters for the Test Bed of Simulated Factories**

Velocities			Mean Work Content by Machine						
ν_1	ν_2	ν_3	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	Max/Min
5	10	15	3.95	7.50	5.09	5.09	5.16	3.20	2.34
8	10	12	3.15	5.82	6.64	2.50	5.55	6.34	2.66
10	10	10	2.50	6.21	5.49	6.44	4.19	5.17	2.58
8	9	13	4.12	4.03	4.07	7.07	7.50	3.22	2.33
7	11	12	5.68	4.50	3.12	7.50	4.07	5.14	2.40
			6.15	7.13	6.54	4.88	2.50	2.80	2.85
			5.45	6.26	5.00	5.42	2.50	5.37	2.50
			5.25	2.64	7.50	3.73	6.03	4.86	2.84
			4.03	7.50	4.33	4.93	4.46	4.75	1.86
			6.37	5.74	5.81	5.45	2.50	4.13	2.55
			3.59	7.50	4.49	3.36	4.13	6.94	2.23
			5.83	7.50	5.15	4.76	3.91	2.85	2.63
			4.87	4.63	6.55	3.51	7.50	2.93	2.56
			2.90	5.01	2.50	6.72	5.56	7.30	2.92
			6.50	5.95	2.50	3.93	4.37	6.74	2.70
			2.50	6.18	6.09	7.25	3.73	4.25	2.90
			3.63	7.50	2.61	7.25	4.83	4.19	2.87
			3.23	4.60	4.62	5.56	4.50	7.50	2.32
			5.00	2.50	4.55	7.46	6.52	3.97	2.98
			4.58	7.50	5.72	3.18	3.48	5.54	2.36
			7.50	7.03	6.67	3.03	2.66	3.12	2.82
			4.09	5.99	5.41	5.45	2.50	6.56	2.62
			2.50	6.04	5.25	5.14	5.84	5.23	2.42
			3.10	4.19	7.06	6.68	6.47	2.50	2.82
			4.03	6.68	5.65	4.54	6.61	2.50	2.67

mean work content is not uniform since the linear transformation in Equation (4) is different for each factory. The exact shape of this distribution is not important; the method was designed to provide a wide range of values rather than to represent specific situations. The algorithm completely specifies the distribution so that other researchers can reproduce it.

Some of the worker/machine combinations resulted in assignments with a bottleneck. For example, when the first worker's velocity is 5, a bottleneck occurs if the first machine's work content is greater than 5. Bottleneck designs were eliminated from consideration since it would be inefficient to use such a design when alternatives (*e.g.,* other sequences) are available. This eliminated 13 cases for the Slowest-to-Fastest worker sequence and 9 cases for Fastest-to-Slowest. However, there were only 3 cases for which both sequences had a bottleneck. The resulting test bed consists of 122 cases.

3.3. The Simulation

Worksharing was simulated using a computer program written in FORTRAN. Each simulation run ended after 10,000 units were produced, including a run-in of 1,000 units. The run-in began with empty buffers and idle machines and workers. Variance reduction was accomplished in two ways: The same sequence of random numbers was used for each simulation, and each machine has its own random number generator. Thus job j on machine i has the same work content in every run of a given factory. Processing times were obtained by generating random work content for the machine and dividing by the velocity of the worker. If preemption occurs, the remaining processing time is adjusted according to the new worker's velocity.

4. Worksharing with Preemption

Worksharing systems in which workers are allowed to preempt their upstream neighbor are studied in this section. We begin with an analysis of variability and worker-to-machine ratio in low-inventory systems such as bucket brigade. Adding small in-process inventories and using them to control worker movement is considered next. Then the effects of worker sequence, zone size, and controls are studied using a $2 \times 2 \times 4$ design on the test bed described in §3.2. Finally, we analyze the systems that perform poorly to see what hypotheses we can generate for future research.

4.1. Preliminary Analyses (with Preemption)

It is well established that one cause of idle time in serial production facilities is variation in processing time. In worksharing lines the flexibility to use different machines is expected to compensate for processing time variability, but this flexibility is limited if the ratio of machines to workers is low. Since these observations are not controversial, a limited design was used to investigate the magnitude of the effect. (The results are also used as a guide to selecting parameter values used in subsequent analyses.) The systems studied have three workers with velocities 8, 10, and 12, Slowest-to-Fastest order, identical machines, and no inventory buffers.

Figure 3 shows the results for both "Max Zones" (as used in bucket brigade) and "Full Zones" (worker i's zone begins at machine i). As expected, the curves show substantial loss of efficiency for large coefficient of variation and short production lines. Given the limited scope of the systems studied, we list these as "observations" rather than conclusions.

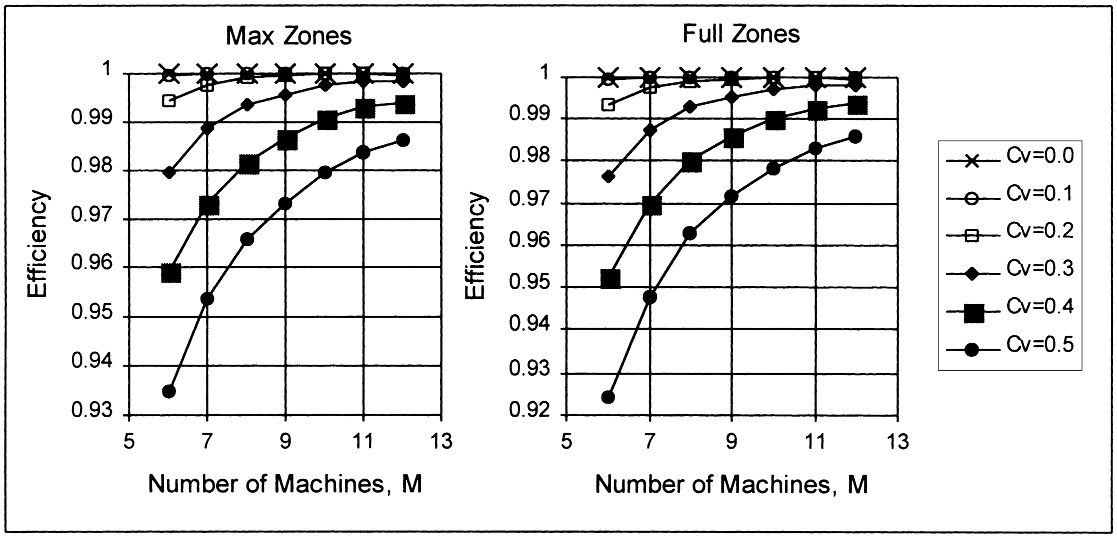
Observation 1. *Variation in processing time causes loss of efficiency in worksharing lines.*

Observation 2. *Efficiency increases as the ratio of machines to workers increases.*

Figure 3 also shows that Max Zones perform very slightly better than Full Zones in this case. We will see later that there are situations where this is not true.

Low efficiency is money lost. In the preceding examples, even modest variability caused lower efficiency for lines of six machines. The literature shows that a very small amount of inventory is usually sufficient to overcome variation in processing time. The bucket brigade system does not allow inventory. Even a worker who overtakes another is not allowed to leave the item and seek work upstream, but is required to wait until the next worker finishes. This kind of delay is more likely as variability increases or if workers have similar velocities, despite the Slowest-to-Fastest sequence.

Figure 3 **The Effect of Variation and Machine-to-Worker Ratio on Efficiency (with Preemption): Identical Machines, Worker Velocities (8, 10, 12)**



In similar environments Ostolaza et al. (1990) showed that inventory controls, targeted at maintaining buffers near half-full, increase efficiency. Half-full provides a safety margin to reduce starving, while half-empty similarly reduces blocking. Actively maintaining inventory at or near the target thereby reduces idle time.

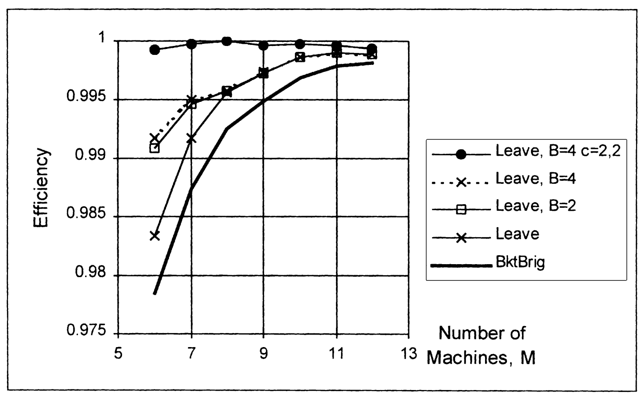
For an initial investigation of inventory policies, CV of processing time was held constant at 0.3, which is near the center of the values reported in §3.1. Max Zones are used to allow a direct comparison to bucket brigade. Control levels were either zero (no control) or equal to half of the buffer capacity. (Section 4.3 contains a more detailed exploration of controls.)

The results are shown in Figure 4. The top line shows that using half-full controls upstream and downstream (c = 2,2) substantially improves efficiency. The remaining four lines demonstrate the extent to which inventory without controls can increase efficiency. Specifically, Leave (workers allowed to leave inventory on a machine) outperforms Bucket Brigade (BktBrig), and additional storage capacity (Leave, B = 2 and Leave, B = 4) causes more improvement. To summarize:

Observation 3. *Efficiency increases when inventory is allowed in worksharing lines with variable processing times.*

Observation 4. *Efficiency increases if inventory targets of half-full buffers are employed to control the sharing of operations.*

Figure 4 **The Effect of Inventory and Controls on Efficiency (with Preemption): Max Zones, Identical Machines, CV = 0.3, Velocities (8, 10, 12)**



People are different. The worker velocity profile (8, 10, 12) used in the previous examples reflects a modest difference, but other examples should conform to the preceding observations. However, the effects of zone size and "overtaking rules" depend importantly on the velocity profile. To illustrate this, we varied the velocity of workers 1 and 3 by as much as 50% in a symmetrical manner, maintaining the Slowest-to-Fastest sequence suggested for Bucket Brigade. That is, low velocity for worker 1 is exactly offset by high velocity for worker 3 so that the theoretical capacity of the line remains constant.

Figures 5a and 5b show the results for lines with six and twelve identical machines, respectively. The next observation is based on the left side of Figure 5a, where all of the curves involving Full Zones show reduced efficiency as the velocity of worker 1 decreases.

Observation 5. *Systems with Full Zones have degraded performance when the first worker's "fair share" consists almost entirely of the first machine.*

The following example illustrates an explanation of this effect. For velocities (5, 10, 15) the first worker's velocity is one-sixth of the total, so one machine is a full load for worker 1 in a six-machine line. However, because of variation in processing time, it is not uncommon for worker 2 to need another item before worker 1 is finished on machine 1. With Full Zones worker 2 remains idle. With Max Zones, worker 2 takes over machine 1 and finishes that unit in half the time that worker 1 would need. Thus Max Zones achieves higher throughput by idling a slower worker instead of a faster one.

The effect disappears for 12 machines (see Figure 5b) because worker 1's one-sixth share is the first two machines. Since the above problem only occurs when worker 1 is using the first machine, it is much less common.

The right side of Figures 5a and 5b suggest another observation, since all of the Bucket Brigade curves drop as worker 1's velocity approaches 10.

Observation 6. *When worker velocities are nearly equal, Bucket Brigade performs poorly even with high machine-to-worker ratios. The Leave rule performs better in these circumstances.*

This happens because it is not unusual for worker 1 to catch up to worker 2 when all workers have the same average velocity. Under Bucket Brigade worker 1 must remain idle until worker 2 finishes. Under the "Leave" rule the overtaking worker abandons the unit and goes back to start a new one, thereby avoiding idle time. Figure 5b makes it clear that the main cause of this problem is the "Wait" rule—removing that rule eliminates the problem for both Max and Full Zones when there are plenty of machines.

One might wonder why this effect persists for the 4-to-1 machine-to-worker ratio of Figure 5b. With $M = 12$ there is a lot of room for workers to find something to do so idle time should be extremely rare. We propose the following possible explanation. Overtaking is less likely with more machines, but when it does happen Bucket Brigade's "Wait" rule keeps the workers together until the operation is complete, increasing the chance that the same problem will happen again. That is, the "Wait" rule tends to cause workers to "bunch up" on the line whereas the "Leave" rule causes them to separate.

Based on the preceding observations, it is not surprising that Max Zones with the "Leave" rule is nearly immune to either of the above effects, as evidenced by its near-horizontal line in both Figure 5a and Figure 5b. Max Zones tends to avoid the "slow first worker problem" and "Leave" avoids "overtaking delays" for equal workers.

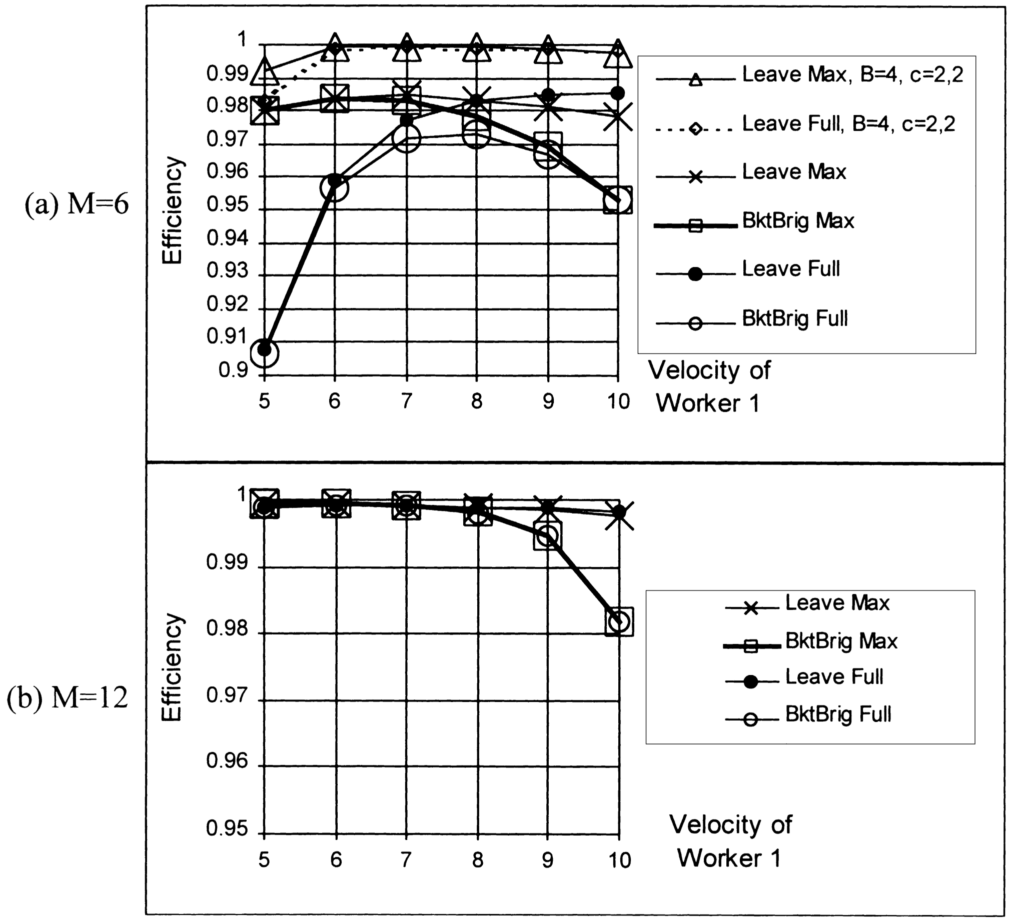
Finally, Figure 5a shows again that adding buffers with inventory controls improves efficiency substantially. For example, in most cases a little inventory, properly managed, provides buffers as effective as six extra machines.

The apparel plant we observed did not want to allow the first worker to be preempted from machine 1, since that would "force him/her off of the line." Hence they operated with "Full" zones rather than "Max" zones. Also, their goal was to keep everyone busy as equally as possible, as a matter of fairness, and allowed small amounts of inventory to avoid idle time. These desiderata guided our choices of parameters in the next section.

4.2. Worker Sequence, Zone Size, and Buffer Management: A 2 × 2 × 4 Design

The objective of this analysis is to determine which combinations of zone size, worker sequence, and controls work effectively, and which combinations cause difficulty. To expand the analysis for these parameters while maintaining a manageable number of cases, we found it necessary to restrict other parameters. Zone size is restricted to "Full" (as in the apparel factory we

Figure 5 **The Effect of Worker Velocity Profile (with Preemption): Max Zones, Identical Machines, CV = 0.3, Velocities (x, 10, 20-x)**



observed) or "Minimal" (the smallest possible for worksharing). Max Zones are excluded, but given the use of buffers (see below) the results should be similar to Full Zones, as in Figure 5a.

Coefficient of variation is held constant at 0.3, a figure that is in the middle of the ranges reported in the literature (see §2). For the uniform distribution, CV = 0.3 corresponds to a range from approximately half of the mean to 1.5 times the mean. For the normal distribution with the same CV, about 10% of observations will be outside that range.

Figure 3 showed that lines of six machines have substantial loss of efficiency for CV = 0.3, so all of the runs in this section have six machines. The smallest worksharing system in which a worker has both upstream and downstream neighbors has 3 workers, so that value was chosen. A six-machine 3-worker line is at the bottom of the observed range of machine-to-worker ratios cited in §3. Hence the values chosen represent a system that could be encountered if machines were relatively expensive compared to most cases cited in the literature.

Figures 4 and 5 (and other literature cited in § 2) suggest that buffer capacity of 4 with suitable controls may be sufficient to restore nearly 100% efficiency, given all of the above characteristics, so that value was chosen for the control buffers in this experiment.

The remaining two factors are worker sequence (Slowest-to-Fastest or Fastest-to-Slowest) and use of inventory controls to govern the sharing of tasks, with four choices: none $c = (0,0)$; downstream $c = (0,2)$; upstream $c = (2,0)$; or both, also called *dual controls c*

Table 2 Worker Sequence, Zone Size, and Buffer Management (with Preemption)

		Slowest-to-Fastest, $n = 16$		Fastest-to-Slowest, $n = 112$	
Controls:		Min Zones (Smin)	Full Zones (Sfull)	Min Zones (Fmin)	Full Zones (Ffull)
None	Average efficiency	0.9746	0.9855	0.9738	0.9813
$c = 0,0$	Lowest efficiency:	0.9131	0.9108	0.8716	0.8850
	# of times <0.99	82	62	79	66
Downstream	Average efficiency:	0.9889	0.9934	0.9898	0.9898
$c = 0,2$	Lowest efficiency:	0.9364	0.9665	0.8530	0.8980
	# of times <0.99	39	27	23	26
Upstream	Average efficiency:	0.9630	0.9820	0.9693	0.9840
$c = 2,0$	Lowest efficiency:	0.8977	0.9081	0.9039	0.9468
	# of times <0.99	97	77	96	77
Dual	Average efficiency:	0.9956	0.9928	0.9946	0.9947
$c = 2,2$	Lowest efficiency:	0.9736	0.9604	0.8702	0.9605
	# of times <0.99	14	27	12	17
Performance of Best* of 16					
	Average efficiency:	0.9984			
	Lowest efficiency:	0.9934			
	# of times <0.99	0			

*"Best" assumes that all designs are tried and the one with highest efficiency is selected.

= (2,2). Note that when controls are used, they have targets of half-full, as discussed earlier.

In discussing the results we use notation as follows: Sfull(0,2) indicates a system with *Slowest-to-Fastest* worker order, *Full* Zones, and downstream controls only. The test bed of 122 factories described in §3.2 was used to compare all combinations of the design parameters. (See §2.3 for definitions and §3.3 for information about the simulations.)

Based on Bartholdi and Eisenstein (1996) we expect Slowest-to-Fastest to be preferred. Further, we expect Full Zones to be preferred over Minimal Zones because larger zones should act as a form of buffer, giving workers more ways to avoid becoming idle. Finally, we expect downstream controls to be more important than upstream because preemption makes inventory less necessary at the upstream position. For example, downstream control keeps worker 1 near the front of the line until some inventory is in place that either worker 1 or worker 2 may use on a shared machine. Once that is done, worker 1 may proceed downstream but can be preempted when worker 2 needs a new item. In contrast, upstream control causes worker 2 to build inventory *after* the ideal handoff machine, which is fine for that worker but is of less use for worker 1 since that inventory's next processing step is on a machine that worker 2 must use most of the time if the line is to be balanced. (A somewhat different expectation will be given in § 5 in which preemption is not allowed.)

Table 2 shows simulation results, summarized using three measures: average efficiency, lowest efficiency, and number of factories for which the given design did not achieve at least 99% efficiency. The results show clearly that *no system is best in every case!* However, there are many inferences we can draw. After a brief discussion of the table we present ANOVA results.

The best system in terms of average throughput is Smin(2,2), which is Slowest-to-Fastest worker sequence, Minimal Zones and dual controls or $c = (2,2)$. This is consistent with our expectations that Slowest-to-Fastest is good and half-full controls increase efficiency. However, it is contrary to our expectation that larger zones act as buffers to increase efficiency. Smin(2,2) has the best worst-case performance, measured here as the lowest efficiency case.

However, using another criterion a different system is best. Fmin(2,2) had fewer instances with efficiency below 99%. Since this system has the opposite worker sequence, it shows that there are specific cases in which Slowest-to-Fastest is not best.

Finally, the summary at the bottom of Table 2 shows that, for every case in the test bed, at least one of the 16 combinations of rules achieves 99% efficiency or higher.

Our expectation that downstream controls are more important than upstream is supported by the data. In fact, upstream controls by themselves seem to do more harm than good on all of the measures in Table 2. As predicted, downstream controls do improve efficiency, but dual controls provide substantial additional improvement in every case but one.

Analysis of variance (ANOVA) was used to explore the simulation data in greater detail. The analysis included all seven two-way interactions and was carried out using the following dummy (indicator) variables in a linear regression:

Worker Sequence:	SF = 1 if Slowest-to-Fastest, 0 if Fastest-to-Slowest.
Zone Size:	FULL = 1 if Full Zone is used, 0 if Minimal Zone is used.
Controls:	Down = 1 for single, downstream controls (e.g. c = 0,2), 0 otherwise.
	UP = 1 for single, upstream controls (e.g. c = 2,0), 0 otherwise.
	DUAL = 1 for both upstream and downstream controls (e.g. c = 2,2), 0 otherwise.

The base system for comparison (in which all dummy variables are zero) is Fmin(0,0), or Fastest-to-Slowest worker sequence, Minimal Zones, and no controls. Table 3 shows the results. (Note: the same Main Effects were significant whether or not the Interactions were included.)

Worker order (represented by SF) is not statistically significant, using 0.05 as the allowable type I error probability. This is contrary to the expectation based on the Bucket Brigade model. In contrast, zone size (represented by FULL) is significant. As predicted, Full Zones is better on average, despite the fact that the system with the highest average efficiency used Minimal Zones. Buffer control is also significant and consistent with predictions—dual controls are best, downstream controls are good, and upstream controls should not be used by themselves.

Four of the interaction terms are significant. Two of them are interactions with upstream controls, which are (as noted above) not recommended. The other two indicate that the effect of Full Zones is negatively moderated when either Downstream or Dual controls are in place. In fact the *net* effect of Full Zones is negative when Dual controls are used (0.00802–0.01055). This interaction is also evident in Table 2, in which Minimal Zones are best with Dual Controls, but Full Zones are better in the other cases. It also explains the apparent contradiction that Full Zones are better on average, but the best system has Minimal Zones and Dual Controls.

4.3. Analysis of Cases with Relatively Low Efficiency

It is disappointing that a single design did not emerge as a champion, best in all circumstances. However, we have some opinions, based on research only partially reported here, about how to tell when relatively low efficiency will occur and how to fix that. First, as identified in §4.1, efficiency may suffer when machine 1 is a large fraction of the capacity of the first worker. To see whether this problem is a general one, a dummy variable was added to the regression model to indicate designs in which the ratio of average work content for machine 1 to velocity of worker 1 exceeded 0.9. (This value was chosen arbitrarily, based on the results shown in Figure 5a.) The resulting coefficient was negative and significant (coefficient $= -0.01520$, $p = 0.00000$). Hence this effect, when present, can be expected to lower efficiency by 1.5%, ceteris paribus. Including the new variable did not change the signs and significances of the other variables.

Second, we noticed that performance degrades when the slowest worker has an Ideal Handoff Point at a machine with significantly more work content than the average of 5.0. Competition for that machine often leads to blocking or starving. A dummy variable was

Table 3 Dummy Variable Regression (with Preemption)

$R^2 = 0.25883$ Adjusted $R^2 = 0.25392$ $F = 52.704$ P-value = 0.00000

	Coefficient	Standard Error	*P*-value		Coefficients	Standard Error	*P*-Value
Intercept	0.97354	0.00138	0.00000	SF × FULL	0.00234	0.00153	0.12564
SF	0.00130	0.00171	0.44452	SF × DOWN	−0.00111	0.00216	0.60826
FULL	0.00802	0.00171	0.00000	SF × UP	−0.00664	0.00216	0.00211
DOWN	0.01569	0.00188	0.00000	SF × DUAL	−0.00291	0.00216	0.17719
UP	−0.00472	0.00188	0.01203	FULL × DOWN	−0.00690	0.00216	0.00142
DUAL	0.02238	0.00188	0.00000	FULL × UP	0.00770	0.00216	0.00037
				FULL × DUAL	−0.01055	0.00216	0.00000

created to indicate when an ideal handoff machine that involved the slowest worker had work content exceeding 6.5 (again chosen arbitrarily). The resulting coefficient was significant and negative (coefficient $= -0.00690$, $p = 0.00000$). Presence of this effect can be expected to lower efficiency by 0.7%, ceteris paribus.

Interestingly, adding both of these dummy variables made the SF (Slowest-to-Fastest) main effect statistically significant (coefficient $= 0.00353$, $p = 0.03386$). Thus the prediction that SF is best, based on the results of Bartholdi and Eisenstein (1996), seems to be true if troublesome cases represented by the new dummy variables are avoided. More research is needed to improve the identification of these problems, either by improving the measures or by exploring different values of the cutoffs (0.9 and 6.5) that were used to define the dummy variables.

This has practical implications for operating a worksharing line. If a factory encounters either of these problems, changing the sequence of workers may help. Based on informal experimentation on cases with efficiency below 99%, changing worker sequence was the single most powerful method to increase efficiency. More formally, we evaluated "pairs of designs." For example, when the design rule is to choose the better of Smin(2,2) and Fmin(2,2) average efficiency increased from 0.9956 to 0.9975, the lowest efficiency increased from 0.9736 to 0.9821, and only 4 runs were below 99% efficiency, compared to 14 for Smin(2,2) by itself. Thus, simply reversing the worker sequence got rid of 10 of the 14 worst cases. (Two of the remaining 4 systems had identical workers, so reversing the sequence had no effect. The other two had velocity profiles (5, 10, 15) and (15, 10, 5).)

Finally, and even more anecdotally, we worked on the inventory-based controls for some of these problem cases. Changing the control levels seemed to be efficacious when we followed the simple rule, "if a control buffer is more than half-full on average, lower the control level, and vice versa." That is, change the control level in a direction that would tend to move inventory toward half full. Unfortunately, the number of cases is quite small and the exploration was not systematic, so we report this only as a direction for possible future investigation.

When blocking and starving are evident, a manager has several avenues to consider. One is to use a different worker sequence. However, that will not always work. For example, if the workers are nearly equal changing their order will have little effect. More work is needed to develop and evaluate other options. We will discuss this further in the last section. Next, we analyze worksharing systems in which preemption is not allowed.

5. Worksharing, Preemption Not Allowed

As discussed in §2.2 there are important cases in which preemption is not used, and the dynamics of the system are very different without preemption. This section begins with a brief analysis to indicate how large the difference is, and then the test bed is used for a more in-depth analysis.

5.1. Machine-to-Worker Ratio (No Preemption)

Worksharing systems gain efficiency because workers have the flexibility to use different machines. However, this flexibility is limited if preemption is not allowed, so one would predict lower efficiency. However, it is reasonable to assume that this effect should be negligible when the ratio of machines to workers is high, since the *fraction* of time a worker will be idle while waiting for another to finish will be small.

Figure 6 displays simulation results for four design configurations without preemption, based on a single worker-velocity profile (8,10,12), identical machines, and no buffers. It also includes a model *with* preemption for comparison. All of the "NoPreempt" designs have very low efficiency for short lines, and increase in efficiency as line length increases. *However, convergence is quite slow, indicating that disallowing preemption is a serious matter even with 12 times as many machines as workers.*

Observation 7. *Prohibiting preemption lowers efficiency, even for large ratios of machines to workers.*

Note that convergence is not smooth for two of the lines in Figure 6. This is not statistical variation, but is caused by the changing positions of the ideal handoff points. This issue is explored further at the end of this section.

Handoff delay is a key difference that is introduced when preemption is forbidden. It occurs when one worker receives an item directly from another. Without preemption and without inventory, every handoff involves idle time—the downstream worker waits until the upstream worker finishes (except in the rare case when both workers finish simultaneously). Inventory stored between machines should reduce the number of handoffs, thus eliminating some handoff delays. However, since we have already seen that inventory increases efficiency with preemption, this prediction is not tested separately.

Figure 6 **Effect of Machine-to-Worker Ratio on Efficiency (Without Preemption): Max Zones, No Buffers, Identical Machines, Worker Velocities (8, 10, 12)**

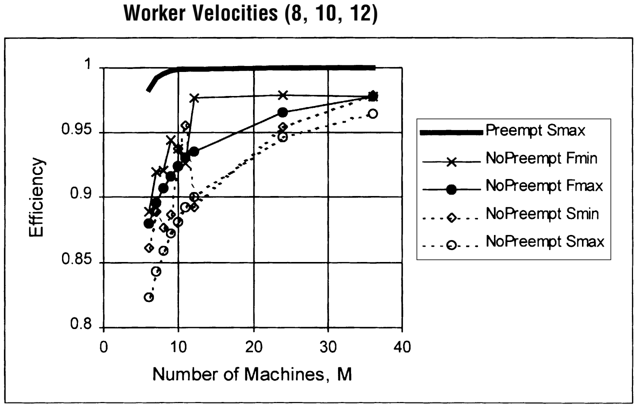


Figure 6 shows some additional features that presage some of the findings of the next section. First, for almost every line length in this example, the Fastest-to-Slowest sequence is better than Slowest-to-Fastest. One reason for this may be that worker sequence affects the *duration* of handoff delay. The faster the upstream worker, the sooner the operation is finished and the shorter the delay. This suggests using the Fastest-to-Slowest worker sequence, the opposite to the one recommended when preemption is allowed.

A second key difference concerns the zone size. In Figure 6 Minimal Zones is better than Full Zones in almost every case. With no preemption allowed, large overlaps between zones can cause trouble. If worker 1 moves deep into worker 2's zone, without preemption worker 2 is blocked from access to much of his/her equipment and may, as a result, suffer idle time.

Figure 6 provides a clear demonstration that inability to preempt may seriously degrade system performance and require different designs. We now turn to a more in-depth analysis.

5.2. Worker Sequence, Zone Size, and Buffer Management (No Preemption)

In this section the $2 \times 2 \times 4$ design described in §4.2 is applied to the NoPreempt situation. The discussion in §5.1 leads us to expect that Fastest-to-Slowest sequence and Minimal Zones will prevail. In addition, the following argument suggests that dual controls will be more important than in the preemption case. Consider an example with three identical machines and 2 identical workers. For the line to be balanced both workers need to use machine M2, and inventory controls are needed on both sides of M2 to avoid handoff delay. When worker 1 is using the first machine, inventory is needed in front of M2 so that worker 2 may use it. When worker 1 uses M2, inventory is needed after M2 to keep worker 2 busy since (s)he is not allowed to take over M2.

Table 4 Worker Sequence, Zone Size, and Buffer Management (No Preemption)

		Slowest-to-Fastest, $n = 116$		Fastest-to-Slowest, $n = 112$	
Controls:		Min Zones (Smin)	Full Zones (Sfull)	Min Zones (Fmin)	Full Zones (Ffull)
None $c = 0,0$	Average efficiency:	0.8356	0.8204	0.9069	0.8987
	Lowest efficiency:	0.6659	0.7275	0.8013	0.8438
	# of times <0.99	116	116	112	112
Downstream $c = 0,2$	Average efficiency:	0.9124	0.9099	0.9582	0.9601
	Lowest efficiency:	0.6865	0.8127	0.8736	0.9113
	# of times <0.99	115	116	100	107
Upstream $c = 2,0$	Average efficiency:	0.8641	0.8653	0.9206	0.9254
	Lowest efficiency	0.7224	0.7372	0.8103	0.8663
	# of times <0.99	116	116	112	112
Dual $c = 2,2$	Average efficiency:	0.9920	0.9643	0.9938	0.9868
	Lowest efficiency:	0.9348	0.8914	0.8818	0.9159
	# of times <0.99	28	100	16	50
Performance of Best* of 16					
	Average efficiency:	0.9963			
	Lowest efficiency:	0.9661			
# of times <0.99	7				

*"Best" assumes that all designs are tried and the one with highest efficiency is selected.

Table 5 Dummy Variable Regression (No Preemption)

$R^2 = 0.69739$ Adjusted $R^2 = 0.69538$ $F = 347.79$ P Value $= 0.00000$

	Coefficient	Standard Error	*P*-value		Coefficients	Standard Error	*P*-Value
Intercept	0.90637	0.00296	0.00000	SF × FULL	−0.00893	0.00326	0.00623
SF	−0.07033	0.00365	0.00000	SF × DOWN	0.02681	0.00461	0.00000
FULL	−0.00720	0.00366	0.04920	SF × UP	0.01643	0.00461	0.00037
DOWN	0.05066	0.00402	0.00000	SF × DUAL	0.06270	0.00461	0.00000
UP	0.01291	0.00402	0.00133	FULL × DOWN	0.01141	0.00461	0.01339
DUAL	0.09042	0.00402	0.00000	FULL × UP	0.01470	0.00461	0.00146
				FULL × DUAL	−0.00580	0.00461	0.20831

Table 4 shows the performance of the designs. All of the preceding conjectures are supported in general terms. The best (highest average efficiency) system is Fmin(2,2), exactly matching the conjectures. Each dual-control system is better than its lesser-control counterparts on every criterion. Fastest-to-Slowest sequences best their counterparts in almost every respect. (The worst-case performance of Fmin(2,2) is beaten by Smin(2,2).) Minimal Zones are better in more than half of the cases, but there are plenty of exceptions.

Table 5 gives results of the same regression model used for the Preempt case. The main effects confirm our expectations.

- Slowest-to-Fastest (SF) has a negative effect, predicting 7% lower efficiency than Fastest-to-Slowest, ceteris paribus.
- Downstream controls have a strong positive effect, but the effect of dual controls is nearly twice as large, adding over 9% to efficiency, far higher than in the preemption case.

- Full Zones are slightly worse than Minimal Zones, but the predicted difference is only -0.7%.

The negative interaction between Full Zones and Slowest-to-Fastest sequence (SF $\times$ FULL $= -10.9\%$) more than doubles the already-negative main effect of Full Zones. That is, Slowest-to-Fastest and Full Zones are a very bad combination.

NoPreempt and Preempt share causes of poor performance. As with the Preempt case, dummy variables were created to indicate a high ratio of work content of machine 1 to velocity of worker 1 and an ideal handoff point involving the slowest worker and a machine with high work content. Both variables were significant and negative, and their presence did not change any other coefficients in either sign or significance. Their coefficients are -0.02729 and -0.00578, respectively.

Pairing the best system Fmin(2,2) with its reverse-sequence twin Smin(2,2) improved average efficiency from 0.9920 to 0.9961, worst-case from 0.9348 to 0.9583, and cases below 99% efficiency from 28 to 8. Although the last two represent substantial improvements, there are still more problem cases with lower efficiency than with preemption allowed. This (and other anecdotal explorations, not presented here) suggests that it may be more difficult to improve efficiency when preemption is not allowed.

6. Conclusions

Worksharing is a powerful tool for increasing efficiency by avoiding worker idle time. It has previously been established that using overlapping zones in serial lines can essentially eliminate idle time in some cases. However, we have shown that there are many realistic situations where more is required. In situations with low machine-to-worker ratios, machine work contents that are quite different from each other, and/or substantial random variation (perhaps due to having a mixed-model line), care must be taken to design a system that avoids idle time. This paper adds to the understanding of worksharing in serial lines by identifying situations where difficulties arise, testing alternatives to help understand the sources of the problems, and presenting alternatives to alleviate them.

Generalizing beyond the scenarios used in the analyses is always an issue. Our observations were based on situations that are similar to those explored or reported in practice by other researchers, so there is reason to expect that many of the properties noted here will be useful in designing systems in similar environments. However, many of our experiments used situations chosen because they are difficult to operate efficiently. Nevertheless, our conclusions are supported by explanations based on the nature of the worksharing environment and rules. The reader may examine our explanations to see whether they apply in a particular situation. In that spirit we offer the following summary.

Bucket Brigade systems, known to work well in low-variability unequal-worker environments, were shown to suffer loss of efficiency in some examples with random processing times and nearly-equal workers. Bucket Brigade uses differences in worker velocity to avoid congestion—putting the slowest worker first prevents workers from overtaking one another. However, if variability is large and worker velocities are nearly equal, substantial idle time occurs because workers catch up to their downstream neighbors and have to wait for the next worker. We have demonstrated that relaxing Bucket Brigade's "wait" restriction can increase efficiency substantially under those circumstances, and that small amounts of inventory can yield further increases.

A second insight came from observing the slightly-different Full Zone and Max Zone designs. (For worker i, the Full Zone begins at machine i, the Max Zone at machine 1.) Full Zones run into trouble when the first worker is very slow. When a long processing time makes machine 1 a bottleneck, Max Zones allow the second worker to preempt, thus "forcing the first worker off of the line." Efficiency increases because the second worker is faster. We demonstrated this effect in a variety of circumstances and showed that it can be predicted based on the ratio of the first machine's work content to the first worker's velocity. Hence a manager who prefers Full Zones may be able to avoid this problem by not assigning too many tasks to the first machine.

Having more machines than workers is required for the type of serial worksharing studied here. Increasing the machine-to-worker ratio improves efficiency by giving workers more opportunity to keep busy without interfering with each other. Not surprisingly, this is more important with high variability where blocking and starving are more likely. What is surprising is how slowly the effect occurs when preemption is not allowed (Figure 6).

We have shown that control buffers effectively guide the workers in their use of shared machines. The work rule tested in this paper is quite simple: Just observe whether the storage area between machines is above or below half full. High inventory directs the worker downstream, and low inventory sends her/him upstream.

One of the most important findings is that serial worksharing lines can operate at high efficiency with or without preemption. However, substantial differences occur. (1) Sequencing workers from Slowest-to-Fastest is a good starting point with preemption because it tends to keep the fastest worker busy all of the time. If preemption is not allowed, the opposite sequence is a better starting point because it helps to reduce handoff delay and prevents the fastest worker from being starved. (2) Based on average performance, Minimal Zones are best with or without preemption. However, giving workers more freedom of movement (Full Zones) also works well in many cases when preemption is allowed, but is inadvisable if preemption is prohibited. Without preemption, a worker who moves too far into other zones will often block other workers from reaching some of their designated machines. (3) There are many cases in which systems with preemption operate well with only downstream controls, or even no controls at all (Table 2). Without preemption, both upstream and downstream controls are needed in almost every case (Table 4).

One objective was to find designs that reduce or eliminate idle time in a variety of circumstances. Because of the major differences noted above, we proposed two starting-point designs—one for preemption and one to use when preemption is prohibited. Also, since those designs were not always best, we considered alternative designs and presented evidence that simply reversing the worker sequence often raises efficiency when the original design does not perform well. This observation coincides with the fact that two measures of machine-worker mismatch proved to be significant predictors of efficiency loss. (One measure, mentioned above, indicates a problem on the first machine, and the other indicates a problem on a shared machine.) Changing the sequence of workers may avoid such a mismatch. Additional research could further clarify this connection and lead to improved diagnosis and treatment of low-efficiency lines.

However, in some cases changing the worker sequence makes little or no difference. For example, sequence is irrelevant for equally-skilled workers. In our experience (not documented because of its anecdotal nature) most cases where resequencing did not achieve high efficiency can be substantially improved by changing the control rules using a simple criterion: *If inventory in a control buffer is above half-full on average, lower its control level*, and vice-versa. Thus, serial worksharing seems to be another case for which "a half-full buffer is a joy forever" (Ostolaza et al. 1990). Further work on this issue could help managers decide when to use controls, whether single or dual controls should be implemented, and when to use control levels other than half-full.

To design a serial worksharing system requires information about the jobs to be done and the capabilities of the workers. We have assumed that the partition of the jobs into operations has been given, and that operations have been uniquely assigned to machines. If either of these can be changed, doing so would be another tool to alleviate problems such as "large machine at an overlap." It is also possible that the velocities of the workers are inaccurate, unknown, or dynamic due to learning. In that case, design may require some trial and error because accurately locating the ideal handoff machines (and hence control buffers) may be difficult. One useful research avenue would be to identify designs that are robust with respect to errors in estimation of worker characteristics.

The behavioral effects of low inventory systems are not well understood. Juran et al. (1997) showed that personality of the workers affects productivity of the team. Theories of job design would generally predict higher productivity with worksharing due to higher interdependence (Guzzo and Shea 1992), greater task variety and identity (Hackman 1987), more flexibility (Goodman 1979), and smaller teams (Steiner 1972). For a good review of how these constructs can affect productivity see Campion et al. (1993). Social loafing theory could predict either an increase or a decrease in productivity depending on whether worker identification is decreased due to sharing of tasks or heightened by the greater interaction among workers (Williams et al. 1981). Schultz et al. (1998) demonstrated that productivity can increase in low-inventory situations, and that this change is attributable to feedback and the development of group norms. Simulation models that do not take this into account can substantially underestimate the productivity of the line. More research is needed to better understand these issues in a variety of environments.[1]

[1]The authors would like to acknowledge the assistance of the Editor-in-Chief, Senior Editor, and referees; their work improved this paper significantly.

References

Bartholdi, John J. III, Donald D. Eisenstein. 1996. A production line that balances itself. *Oper. Res.* **44** (1) 21–34.

——, ——, Charlotte Jacobs-Blecha, H. Donald Ratliff. 1995. Design of bucket brigade production lines. Working Paper. University of Chicago, Chicago, IL.

Berg, P., E. Appelbaum, T. Bailey, A. L. Kalleberg. 1996. The performance effects of modular production in the apparel industry. *Indust. Relations* **35**(3) 356–373.

Bischak, Diane P. 1996. Performance of a manufacturing module with moving workers. *IIE Trans.* **28**(9) 723–733.

Campion, Michael A., Gina J. Medsker, A. Catherine Higgs. 1993. Relations between work group characteristics and effectiveness: implications for designing effective work groups. *Personnel Psych.* **46**(4) 823–850.

Conway, R., W. Maxwell, J. O. McClain, L. J. Thomas. 1988. The role of WIP in serial production lines. *Oper. Res.* **36**(2) 229–241.

Goodman, P. S. (with major contributions by Edward Conlon, Dennis Epple, Eduard Fidler). 1979. *Assessing Organizational Change: The Rushton Quality of Work Experiment.* Wiley, New York.

Guzzo, R. A., G. P. Shea. 1992. Group performance and intergroup relations in organizations. M. D. Dunnette, L. M. Hough, eds. *Handbook of Industrial and Organizational Psychology, Volume 3,* Consulting Psychologists Press, Palo Alto, CA. 269–313.

Hackman, J. R. 1987. The design of work teams. J. W. Lorsch, ed. *Handbook of Organizational Behavior.* Prentice Hall, Englewood Cliffs, NJ. 315–342.

Hendricks, Kevin B., John O. McClain. 1993. The output process of serial production lines of general machines with finite buffers. *Management Sci.* **39** (10) 1194–1201.

Juran, David C., Kenneth L. Schultz, John O. McClain, Tove H. Hammer, Lee W. Schruben. 1997. Personality and performance in serial worksharing systems. Working Paper 96-04. Johnson Graduate School of Management, Cornell University, Ithaca, NY.

Knott, Kenneth, Roy J. Sury. 1987. A study of work-time distributions on unpaced tasks. *IIE Trans.* **19**(1) 50–55.

McClain, J. O., L. J. Thomas, C. Sox. 1992. On-the-fly line balancing with very little WIP. *Internat. J. Production Econom.* **27** 283–289.

Moore, Lila. 1995. K-products walks fine line of modular production. *Apparel Indust. Magazine* January.

Muth, E. J. 1973. The production rate of a series of work stations with variable service times. *Internat. J. Production Res.* **11**(2) 155–169.

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

Proper, Carl, Rebecca Plattus. 1993. Cupid hits bull's eye with modular system. *Apparel Indust. Magazine* **September.**

Schultz, Kenneth L., David C. Juran, John W. Boudreau, John O. McClain, L. Joseph Thomas. 1998. Modeling and worker motivation in JIT production systems. *Management Sci.* **44**(12).

Steiner, I. D. 1972. *Group Process and Productivity.* Academic Press, New York.

Williams, K., Harkins, S., Latane, B. 1981. Identifiability as a deterrent to social loafing: Two cheering experiments. *J. Personality and Social Psych.* **40** 303–311.

Zavadlav, E., J. O. McClain, L. J. Thomas. 1996. Self buffering, self balancing, self flushing production lines. *Management Sci.* **42**(8) 1151–1164.

The consulting Senior Editor for this manuscript was Michael Magazine. This manuscript was received on July 14, 1997, and was with the authors 11 months for 4 revisions. The average review cycle time was 55.3 days.