Manufacturing & Service Operations Management

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

A Comparison of U-Line and Straight-Line Performances Under Stochastic Task Times

Suresh Chand, Ting Zeng,

A Comparison of U-Line and Straight-Line Performances Under Stochastic Task Times

Suresh Chand • Ting Zeng
Krannert Graduate School of Management, Purdue University, West Lafayette, Indiana 47906
IMC, Inc., 45625 Willow Pond Plaza, Sterling, Virginia 20164

Traditionally, assembly lines are laid out in a straight-line configuration where a worker covers only adjacent stations. In a U-line layout, on the other hand, two or more nonadjacent stations can be physically close to each other, making it possible for a worker to cover nonadjacent stations. This added flexibility increases the decision space for U-line layouts and can result in better balanced lines. This paper examines the impact of stochastic task times on the relative performance of U-line and straight-line layouts. Several analytical and simulation results are presented, and insights are provided to explain the difference in the performance of U-line and straight-line layouts. To summarize our main results, although balanced U-line layouts are at least as productive as balanced straight-line layouts given deterministic task times, they can be less productive given stochastic task times if they are balanced deterministically using mean times.
(*Assembly Line; U-Line; Stochastic Times; Throughput; Simulation*)

1. Introduction

This paper considers an unbuffered, multistation, worker-paced serial assembly line with stochastic task times and fewer workers than stations. The tasks for the product to be assembled on the line may have precedence relations among them. We define *station i* as the location that performs task *i*; station *i* has the necessary tools and components to perform task *i*. Each worker may be assigned to one or more stations. We define *work area* for a worker as the space that consists of the stations assigned to that worker and where the worker performs the assigned tasks. Units move from one station to the next only after the preceding station has completely finished its task. Consequently, the time between two successive units coming off the line is stochastic. In this paper, we focus on two performance measures: (1) the expected production rate, that is, the reciprocal of the expected cycle time, and (2) the variance of the cycle time. To simplify the presentation, we will use *output rate* to denote the expected production rate and *variability* to denote the variance of the cycle time. Also, we will use *productivity* to denote the combined performance on both the output rate and variability measures.

Traditionally, lines are laid out in a straight-line (SL) configuration where the work area for a worker covers only adjacent stations. In a U-line (UL) layout, on the other hand, two or more nonadjacent stations can be physically close to each other, making it possible for a worker to cover nonadjacent stations. Hence, the UL layout can increase the decision space in designing a line (i.e., in assigning tasks to work areas such that precedence constraints are satisfied and the output rate is maximized). It should be noted that maximizing the output rate, given deterministic task times, requires assigning tasks to work areas in a way that the difference between the maximum-timed workload and the minimum-timed workload is minimized. Thus, given deterministic times, UL layouts can result in better "balanced" lines. Other advantages of UL layouts include

Manufacturing & Service Operations Management © 2001 INFORMS
Vol. 3, No. 2, Spring 2001, pp. 138–150

1523-4614/01/0302/0138$05.00
1526-5498 electronic ISSN

improved visibility, communication, and teamwork (Miltenburg and Wijngaard 1994).

Machine-paced SL layouts have been studied extensively in the literature for both deterministic and stochastic task times; see Talbot et al. (1986) and Baybars (1986) for deterministic times, and Kao (1976) and Carraway (1989) for stochastic times. For UL layouts with deterministic times, Miltenburg and Wijngaard (1994) and Urban (1998) provide algorithms to find optimal solutions for smaller problems and approximate solutions for larger problems.

The U-line layout that we consider is similar to the ''moving-worker'' production line in Bartholdi and Eisenstein (1996), Bischak (1996), and Bartholdi et al. (1999) in the sense that we also have fewer workers than stations and that the WIP inventory in the system does not exceed the number of workers. However, there are important differences. They require that a worker at a station can move only to an adjacent station on the production line, while we allow the workers to move to nonadjacent stations. Also, at least in theory, they require workers to be cross-trained on all stations, while we require workers to be cross-trained only on the stations in their respective work areas. They show that the production lines with workers who are cross-trained on all stations can be self-balancing if management needs to adjust the production volume by changing the number of workers.

The U-line layout in this paper can also be viewed as a ''re-entrant'' flow line because it has some work areas with multiple visits by a unit. There is a rich body of literature on scheduling jobs in semiconductor manufacturing plants with re-entrant flows; see Kumar (1993) for a description of the problem, a brief literature review, and several analytical results. Kumar considers buffered lines with the objective of finding scheduling policies that improve both the mean and the standard deviation of flow times for given stochastic arrivals at the first task. Our model is different in that we limit the WIP inventory to the number of workers, and our objective is to maximize the output rate assuming that there is an unlimited supply of raw materials.

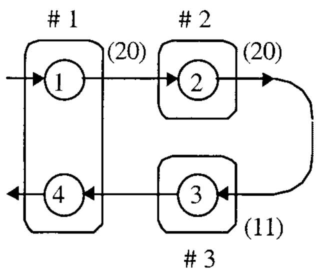
The performance of a line on one or both measures (output rate and variability) declines with randomness in task times; i.e., there is loss of productivity with randomness in task times. This paper focuses on comparing the loss of productivity due to task-time randomness in SL and UL layouts. Specifically, we consider an SL layout and a UL layout that are optimal for a given product with deterministic times. Call these layout designs SL* and UL*, respectively. We know that UL* should be at least as productive as SL* given deterministic times. Is it necessary that the output rate for UL* remains greater than or equal to that of SL* even when task times become stochastic? How does variability in cycle time for UL* compare with that for SL* under stochastic task times? We attempt to answer these questions by providing both analytical and simulation results. Zavadlav et al. (1996) provide some preliminary simulation results to show ''balancing the expected value of work can lead to bad decisions if there is substantial variability.'' To summarize the main results in our paper, we find that while UL* is at least as productive as SL*, given deterministic times, it can become less productive than SL* when task times become stochastic. We identify conditions such that UL* and SL* give the same output rate for stochastic task times. One interesting result is that the performance of UL* depends significantly on the work rules (described in the next section) for the workers whose work areas consist of nonadjacent stations.

The paper is organized as follows. Section 2 of the paper provides computational results for the effect of work rules on the performance of UL layouts. Sections 3 develops several analytical results. It is shown that under certain conditions, UL and SL layouts give the same output rate. However, the variability can be different. Section 4 provides some simulation results. The paper closes in §5 by providing a summary and ideas for further research. Some proofs and a few additional results are provided in the appendices.

2. Work Rules for Unit Selection and Their Effects

Consider the three-worker UL layout for a four-task product (tasks 1, 2, 3, and 4 with expected task times 10, 20, 11, and 10, respectively) shown in Figure 1.

Figure 1 **A UL Layout**



The numbers in parentheses are the expected workloads per unit of production in different work areas.

Suppose both stations 1 and 4 in work area #1 are empty. In deciding what to work on next, the operator for work area #1 may have to choose between two different units. She can either pull a unit of raw material and start working on it at station 1, or she can pull the unit from station 3 (if there is one) and start working on it at station 4. In this example, work area #1 has two entrances (one from raw material and the other one from station 3) and, hence, can have up to two different units available to process whenever she becomes idle. In general, a work area with n entrances can have up to n different units available to process. The work area can process only one unit at a time and needs a work rule to select one of the possible n units.

To discuss details of the proposed work rules, we first explain the production-control assumptions in the paper. We assume that a work area can have at most one unit of WIP inventory. We use a ''pull'' production-control policy for both lines, meaning that, as soon as a worker becomes free, she pulls a new unit into a station in her work area from an entrance to it if: (i) the work area does not already have a unit and (ii) the entrance has a finished unit. We also assume an infinite supply of raw material to station 1 and an unlimited demand for the output. The worker becomes idle if she is not able to pull a new unit into her work area.

For a work area with two entrances, we introduce the following two work rules: (i) ''output-first,'' and (ii) ''input-first.'' Looking at Figure 1, the output-first rule means the operator would give priority to task 4, which is on the output side of the line. The input-first rule means the operator would give priority to task 1, which is on the input side of the line. In the output-first rule, the operator will switch over to station 1 after finishing a unit at station 4 if station 3 does not have a unit that it has finished processing. Note that, with an infinite supply of raw material, there is always work for station 1. In the input-first rule, the operator will switch over to station 4 after finishing a unit at station 1 if both stations 2 and 3 have units that they have finished processing. In this case, the unit at station 3 moves to station 4, while at the same time, the units at stations 1 and 2 move to stations 2 and 3, respectively. We will denote the UL with output-first as UL(O) and the UL with input-first as UL(I).

The input-first rule introduced above can be shown to follow from the ''workload-balancing'' rule in Harrison and Wein (1990) for the situation where the U-line has only two work areas and three stations. Harrison and Wein consider a closed-queuing network with two single-server stations and multiple customer classes, each with its own deterministic route. Each class requires service at specified stations and general service-time distributions. In their simulation study for a line with two servers, the workload-balancing rule had a higher output rate than the shortest-expected-processing-time rule, the shortest-expected-remaining-processing-time rule, and the shortest-next-queue rule.

To study the relative performance of UL(I) and UL(O) work rules, we conducted a simulation study for the layout in Figure 1. Times for tasks 1, 2, 3, and 4 were uniformly distributed with means of 10, 20, 11, and 10, respectively. We varied the coefficient of variation γ (γ = standard deviation/mean process time) to values between 0.0 and 0.5 in steps of 0.1. We used the same γ value for all tasks. We conducted five independent simulation runs for each work rule and each γ. Each run was for 200,000 time periods. The first 100,000 time periods made up the ''run-in'' time; that is, the results for the first 100,000 time periods were ignored. The average cycle times are given in Table 1. Note that cycle times for UL(I) and UL(O) both go up as the task-time randomness goes up. The

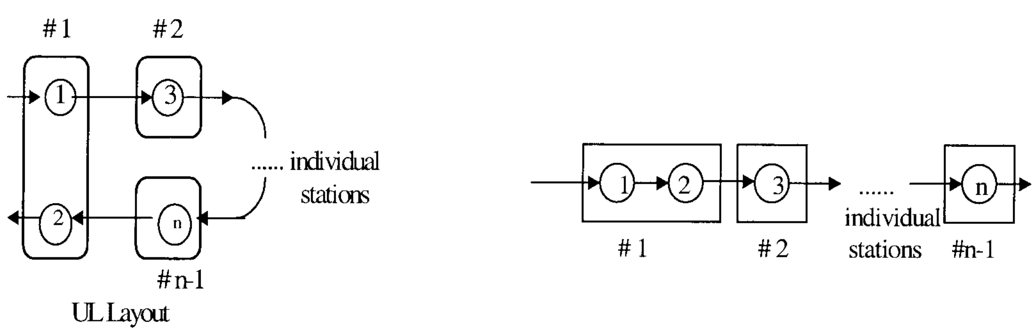
Table 1 Average Cycle Times for the Layout in Figure 1

Coefficient	UL(O)	UL(I)
γ	Expected Cycle Time	Expected Cycle Time
0.0	20.00	20.00
0.1	22.80	21.02
0.2	23.81	21.98
0.3	24.75	23.02
0.4	25.62	24.10
0.5	26.60	25.24

cycle time for UL(I) is lower than that for UL(O) at the 0.05 significance level for all γ values, except for $\gamma = 0$. We also calculated the average WIP inventory for UL(I) and UL(O) for different γ values. On average, the WIP inventory for UL(I) was 7.3% more than that for UL(O). This extra inventory helps reduce the cycle time (i.e., increase the output rate) for UL(I) by reducing starvation.

The next section provides several analytical insights for UL and SL layouts under different operating conditions. We consider both UL(I) and UL(O) for UL layouts. For SL layouts, we consider two different ways of operation: synchronous or asynchronous. An SL layout is synchronous (denoted as SL(syn)) if each unit is held at its work area until all units are ready to proceed to their next work areas. The work area taking the largest amount of time determines the cycle time for a synchronous SL. An SL layout is asynchronous (denoted as SL(asyn)) if the completed unit from a work area proceeds as soon as the next work area becomes free. From Muth (1979), it is known that the output rate for SL(asyn) is greater than or equal to that for SL(syn).

Figure 2 Layouts for a Product with *n* Tasks



3. Analytical Results

We provide these results by considering two layouts with n tasks (Figure 2). While these two layouts are identical on the assignment of tasks to work areas, work area #1 in the UL layout has two entrances and two exits. As a result, while worker 1 in the UL layout can become idle between tasks 1 and 2, worker 1 in the SL layout cannot. In other words, there are more potential block and starve points in the UL layout. This difference in the SL and UL layouts accounts for the difference in the performance of these layouts.

Cycle-Time Results for Deterministic Task Times

First, consider the performance of these systems when task times are deterministic. Let t_i denote the time of task i. Also, let $CT_{\mathrm{SL(syn)}}$ denote the cycle time for SL(syn). $CT_{\mathrm{SL(asyn)}}$, $CT_{\mathrm{UL(O)}}$, and $CT_{\mathrm{UL(I)}}$, are similarly defined. We can now state the following results.

RESULT 1.

$$\left.\begin{array}{ll}\text{(a)} & CT_{\mathrm{SL(syn)}}\\ \text{(b)} & CT_{\mathrm{SL(asyn)}}\\ \text{(c)} & CT_{\mathrm{UL(I)}}\end{array}\right\} = \max(t_1 + t_2, t_3, \ldots, t_n).$$

See Baker et al. (1990) for a discussion of (a) and (b). A proof for (c) is provided in Appendix 2. We next present a result for UL(O) for $n = 3$.

RESULT 2.

$$CT_{\mathrm{UL(O)}} = \begin{cases} \max(t_1 + t_2, t_3) \\ \quad \text{if } t_2 < t_3, \\ \max(t_1 + t_2, t_3) + 0.5 \max(0, t_3 - t_1) \\ \quad \text{if } t_2 \geq t_3. \end{cases}$$

See Appendix 2 for a proof. Given deterministic task times, Result 1 indicates that the UL(I), SL(syn), and SL(asyn) layouts in Figure 2 have an output rate that is equal to the capacity of the work area with the maximum-timed workload per unit of production (or the bottleneck work area). There is no loss of output rate in these systems for deterministic times. However, Result 2 indicates that the use of the output-first rule can cause loss of output rate. We next consider the case with stochastic task times.

Cycle-Time Results for Stochastic Task Times

We again consider the n-task product and the layouts in Figure 2. Let S_i be the random variable for time of task i. The cycle lengths are now random; we let CT_x denote the expected cycle time for layout x.

RESULT 3.

(a) $CT_{\mathrm{SL(syn)}} = \mathrm{E}(\max(S_1 + S_2, S_3, \ldots, S_n))$;

(b) $CT_{\mathrm{SL(asyn)}} = \mathrm{E}(\max(S_1 + S_2, S_3))$ for $n = 3$,

and $\leq \mathrm{E}(\max(S_1 + S_2, S_3, \ldots, S_n))$ for $n > 3$;

(c) $CT_{\mathrm{UL(I)}} = \mathrm{E}(\max(S_1 + S_2, S_3, \ldots, S_n))$;

(d) $CT_{\mathrm{UL(O)}} \geq \mathrm{E}(\max(S_1 + S_2, S_3))$ for $n = 3$.

See Appendix 2 for a brief comment on a proof. Because we have not made any distributional assumptions, the above result holds for any task-time distributions.

Let $CT_{\mathrm{UL(O)}} = \mathrm{E}_1$ and $CT_{\mathrm{UL(I)}} = CT_{\mathrm{SL(syn)}} = CT_{\mathrm{SL(asyn)}} = \mathrm{E}_2$ for $n = 3$ in Figure 2. It is possible to get an expression for $\mathrm{E}_1 - \mathrm{E}_2$ for exponential task times. As discussed in Hopp and Spearman (1996), this is a case of "practical worst-case performance" and is of interest because "any system with worse behavior is a target for improvement." Let $1/\lambda_i$ be the mean of S_i; then it is possible to show:

PROPOSITION. *$E_1 - E_2 = \lambda_3\lambda_1/[(\lambda_3 + \lambda_2)(\lambda_3 + \lambda_1)(2\lambda_3 + \lambda_2)]$.*

See Appendix 1 for a proof. We next provide analytical results for variability.

Variability Results for Stochastic Task Times

Let Y_x denote the random length of the cycle time for layout x (x = SL(syn), SL(asyn), UL(I), or UL(O)). Also, let V_x denote the variance of Y_x and $v(S)$ the variance of random variable S.

RESULT 4. In Figure 2,

(a) $V_{\mathrm{SL(syn)}} \leq V_{\mathrm{UL(I)}}$ for $n \geq 3$.

In addition, when $n = 3$,

(b) $V_{\mathrm{SL(syn)}} \leq V_{\mathrm{SL(asyn)}}$.

(c) The relationship ($\gtreqless$) between $V_{\mathrm{SL(asyn)}}$ and $V_{\mathrm{UL(I)}}$ depends on the relative values of the distributional parameters of S_1, S_2, and S_3.

(d) For exponential task times:

(i) $V_{\mathrm{SL(asyn)}} \geq V_{\mathrm{UL(I)}}$.

(ii) $V_{\mathrm{UL(O)}} \geq V_{\mathrm{UL(I)}}$.

(iii) The relationship ($\gtreqless$) between $V_{\mathrm{UL(O)}}$ and $V_{\mathrm{SL(asyn)}}$ depends on the relative values of the distributional parameters of S_1, S_2, and S_3.

A proof is provided in Appendix 2. In proving (c), we show that UL(I) has a higher variance than SL(asyn) if S_3 is large compared to $S_1 + S_2$, and a lower variance if S_3 is small compared to $S_1 + S_2$. (By S_3 large compared to $S_1 + S_2$, we mean we know a priori that the random variable S_3 is surely larger than or equal to the random variable $S_1 + S_2$.) We also make an observation about allocation of variance between stations 1 and 2. Suppose the total variance $v(S_1) + v(S_2)$ is fixed, but it can be allocated to stations 1 and 2 in any manner. Such an allocation of variance does not affect the variance of $Y_{\mathrm{SL(asyn)}}$, but reduces the variance of $Y_{\mathrm{UL(I)}}$ if less variance is allocated to station 2 and more to station 1 in the UL(I).

Note that the relationships in Result 4 (a) and (b) hold for any task-time distributions. Although we prove relationship (ii) in (d) only for exponential task times, our conjecture is that it holds for any distribution of task times. We provide computational results supporting this conjecture in §4.

Table 2 Summary of Results

Expected Cycle Time	Cycle-Time Variability
• $CT_{UL(I)} = CT_{SL(syn)} \geq CT_{SL(asyn)}$	• $V_{SL(syn)} \leq V_{UL(I)}$
• $CT_{UL(I)} \leq CT_{UL(O)}$ for $n = 3$	• $V_{UL(I)} \leq V_{SL(asyn)}$ and $V_{UL(I)} \leq V_{UL(O)}$ for exponential task times and $n = 3$
• $CT_{SL(asyn)} = CT_{SL(syn)}$ for $n = 3$	• $V_{SL(syn)} \leq V_{SL(asyn)}$ for $n = 3$

Table 3 Simulation Results for Task Times 10, 12, 20, and 10

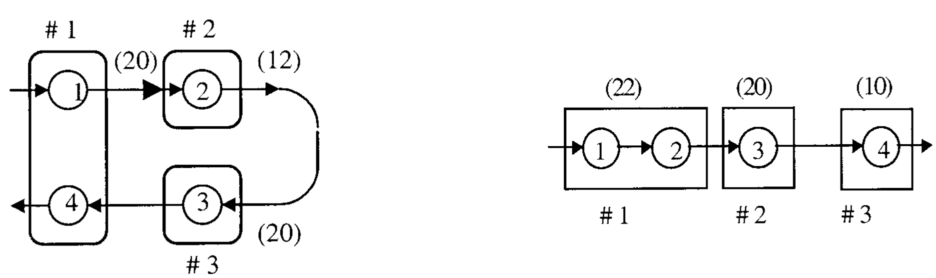
γ	$CT_{UL(I)}$	$V_{UL(I)}$	$CT_{SL(asyn)}$	$V_{SL(asyn)}$	$CT_{SL(syn)}$	$V_{SL(syn)}$
0.0	20.00	0.00	22.00	0.00	22.00	0.00
0.1	21.00	2.97	22.31	10.41	22.31	3.89
0.2	21.98	11.89	22.58	34.28	22.60	15.52
0.3	23.08	26.67	23.54	71.95	23.63	34.84
0.4	24.15	45.28	25.19	126.38	25.19	63.07
0.5	25.31	71.10	26.28	177.05	26.34	95.42

Summary of Results for Stochastic Task Times

The key results for stochastic task times are summarized in Table 2. Comparing SL(asyn) with UL(I), the effect of multiple entrances and exits in UL(I) is to increase the number of points with potential for blocking and starving. As a result, the expected cycle time for UL(I) increases to the level of SL(syn). It is shown that UL(I) has lower variability than SL(asyn) for $n = 3$ and exponential task times. But the variability for UL(I) is not always lower than that of SL(asyn), as seen from (c) in Result 4. Thus, compared to SL(asyn), multiple entrances and exits in UL(I) could induce some synchronization and reduce variability. However, even with this possible reduction, SL(syn) still dominates UL(I) on variability.

The results in Table 2 apply when both SL and UL layouts have identical loads on different work areas, but differ on the number of entrances and exits. We were able to prove analytical results only for identical loads. The next section provides computational results for the case when SL* and UL* do not have identical loads. Note that, as discussed in §1, UL* may have better balanced loads than SL*.

Figure 3 Layouts for a Product with Task Times 10, 12, 20, and 10



4. Computational Results

We conducted a simulation experiment on a 4-task product with 3-worker UL and SL layouts shown in Figure 3. We chose uniformly-distributed task times with expected values of 10, 12, 20, and 10 for tasks 1, 2, 3, and 4. The workloads on three work areas in the UL layout are 20, 12, and 20, which are more balanced than the workloads of 22, 20, and 10 in the SL layout. We conducted simulation experiments similar to those in §2 using the same values of the coefficient of variation. The computational results are given in Table 3; we do not include results for UL(O) as it is dominated by UL(I).

Although the simulated performance of UL(I) dominates both SL(asyn) and SL(syn), it is interesting to analyze the gap between the expected cycle times of UL(I) and SL(syn) as the randomness increases. The expected cycle times for SL(syn) and SL(asyn) are very close to each other. The gaps for different values of γ are shown in Table 4.

It is interesting to note that the cycle-time advantage of UL(I) over SL(syn) does not decrease monotonically with increases in randomness. The cycle-

Table 4 Gaps in Performance

Coefficient of Variation γ	0.0	0.1	0.2	0.3	0.4	0.5
$CT_{SL(syn)} - CT_{UL(I)}$	2.0	1.31	0.62	0.55	1.04	1.03

time advantage of UL(I) over SL(syn) goes down with a small amount of randomness in task times (down from 2.0 to 0.55 for an increase in γ from 0.0 to 0.3). It then goes up and stabilizes. The effect of a small amount of randomness in task times can be explained by considering the workloads in the SL(syn) and UL(I) layouts. With a small amount of randomness, the bottleneck is more likely to shift between work areas #1 and #3 in UL(I) than between #1 and #2 in SL(syn). This is because UL(I) is better balanced than SL(syn). Thus, we expect a larger increase in cycle time for UL(I) than for SL(syn) with small randomness.

When looking at the effect of load balance on variability, we find that the variability for UL(I) is even lower than that for SL(syn). If these systems had identical loads, then, from §3, we would expect the variability for UL(I) to be higher than that for SL(syn). Thus, the effect of better load balance in UL(I) is to reduce variability.

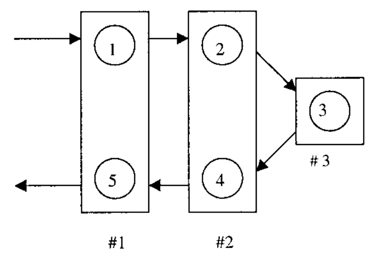
We next provide conclusions and ideas for further research.

5. Conclusions and Ideas for Further Research

We have investigated the relative performances of UL and SL layouts for stochastic task times. Within UL, we considered two work rules: ''input-first'' (UL(I)) and ''output-first'' (UL(O)). Within SL, we considered two types of operations: synchronous (SL(syn)) and asynchronous (SL(asyn)). We considered unbuffered lines with a maximum of one unit of WIP at any work area. Our major findings are:

(1) The performance of a UL layout design is greatly affected by the choice of work rules. Our results for a simple layout in Figure 2 for $n = 3$ show that UL(I) has higher output rate than UL(O). In a more complicated line, it is possible that some mix of input-first and output-first work rules could perform better than any one of these. For example, in the layout given in Figure 4, it may be optimal to use input-first in work area #1 and output-first in work area #2. This question needs to be investigated further. Also, the effect of buffers on the choice of work rules needs to be investigated.

Figure 4 A Five-Station U-Line with Two Crossovers



(2) The effect of better load balance in UL over SL is such that some benefits of UL over SL may decline with a small increase in task-time randomness due to shifting bottlenecks. This factor needs to be considered when choosing between UL and SL layout designs. Another effect of better load balance is to reduce variability of the cycle time.

(3) Multiple entrances and exits in UL(I) can provide better synchronization in UL(I) and, as a result, reduce the variability of the cycle time compared to that in SL(asyn). However, the output rate is likely to go down compared to that in SL(asyn).

(4) We identified situations when both UL(I) and SL(syn) give the same expected cycle time.

(5) Whether or not the reversibility property holds for UL depends on the selection of work rule. (See Appendix 2 for a demonstration of this result.)

Finally, this paper analyzed an optimal balance for UL layout that is found deterministically using the mean task times. A useful question is how to find an optimal balance for UL layout for stochastic task times.

Acknowledgment
Partial funding for this research was provided by the Dauch Center for the Management of Manufacturing Enterprises, Purdue University.

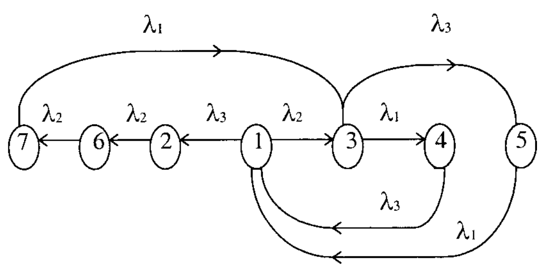
Appendix 1.

We derive expressions for cycle time and variability for UL(O), SL(asyn), and UL(I) in Figure 2 with $n = 3$ and exponential task

Table 5 UL(O) States

State	Station 1	Station 2	Station 3
1	E	W	W
2	E	W	B
3	W	E	W
4	B	E	W
5	W	E	B
6	E	W	E
7	W	E	E

Figure 5 State Transition Diagram for UL(O)



times. Due to exponential task times, Markovian analysis can be used to find the performance measures. We let

S_i: process time of task i with p.d.f. $= \lambda_i e^{-\lambda_i S_i}$, $i = 1, 2, 3$, and
t_i: mean of S_i, $t_i = 1/\lambda_i$.

We use the following notation to define the states

W: working (the station is busy processing a unit)
B: blocked (station has finished the task, but the next station is not ready to pull)
E: empty (station does not have a unit to work on)

The UL(O) layout in Figure 2 with $n = 3$ has seven states (shown in Table 5 and Figure 5). Let p_i represent the long-run portion of visits to state i, $i = \{1, 2, 3, 4, 5, 6, 7\}$; p_i can be obtained by solving the following steady state equations:

$$p_4 + p_5 = p_1, \tag{1}$$

$$p_1 \cdot \lambda_3/(\lambda_2 + \lambda_3) = p_2, \tag{2}$$

$$p_1 \cdot \lambda_2/(\lambda_2 + \lambda_3) + p_7 = p_3, \tag{3}$$

$$p_3 \cdot \lambda_3/(\lambda_1 + \lambda_3) = p_5, \tag{4}$$

$$p_3 \cdot \lambda_1/(\lambda_1 + \lambda_3) = p_4, \tag{5}$$

$$p_6 = p_7, \tag{6}$$

$$p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 = 1. \tag{7}$$

Define the output instant as the moment when a unit has just been completed and exited the line. An output interval is simply the time elapsed between two consecutive instants. The immediate system states after an output instant must be state 7, 3, or 6. If the system is in state 7, the next output interval is $x_7 = S_1 + \max(S_1, S_3) + S_2$, because it takes S_1 time periods to go from state 7 to state 3, and then $\max(S_1, S_3)$ time periods to go from state 3 to state 1 (state 1 is an output state), and it takes S_2 time periods to produce an output. Similarly, if the system is in state 3, the next output interval is $x_3 = \max(S_1, S_3) + S_2$; and if the system is in state 6, the output interval is $x_6 = S_2$. Therefore, the *average output interval or average cycle time* $\mathrm{E}(x)$ can be obtained as:

$$\mathrm{E}(x) = \mathrm{E}(p_7 \cdot x_7 + (p_3 - p_7) \cdot x_3 + p_6 \cdot x_6)/(p_6 + p_3)$$
$$= (p_7 \cdot \mathrm{E}(x_7) + (p_3 - p_7) \cdot \mathrm{E}(x_3) + p_6 \cdot \mathrm{E}(x_6))/(p_6 + p_3), \tag{8}$$

where, using $\mathrm{E}(\max(t_1, t_3)) = 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)$, values of $\mathrm{E}(x_7)$, $\mathrm{E}(x_3)$, and $\mathrm{E}(x_6)$ are

$$\mathrm{E}(x_7) = 1/\lambda_1 + 1/\lambda_2 + 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3),$$

$$\mathrm{E}(x_3) = 1/\lambda_2 + 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3),$$

$$\mathrm{E}(x_6) = 1/\lambda_2.$$

Note that we use $(p_3 - p_7) \cdot x_3$ for state 3 because a portion of transitions into state 3 is from state 7 and the output interval for these is already included in x_7. We divide $\mathrm{E}(p_7 \cdot x_7 + (p_3 - p_7) \cdot x_3 + p_6 \cdot x_6)$ by $(p_6 + (p_3 - p_7) + p_7)$ to normalize the $\mathrm{E}(x)$. The output rate can be expressed as

$$\text{Output Rate} = 1/\mathrm{E}(x). \tag{9}$$

We calculate the variance of the output interval, $\mathrm{Var}(x)$, using $\mathrm{Var}(x) = \mathrm{E}(x^2) - [\mathrm{E}(x)]^2$. $[\mathrm{E}(x)]^2$ can be found similarly to (8). Note that

$$\mathrm{E}(x^2) = (p_7 \cdot \mathrm{E}(x_7^2) + (p_3 - p_7) \cdot \mathrm{E}(x_3^2) + p_6 \cdot \mathrm{E}(x_6^2))/(p_6 + p_3).$$

The terms $\mathrm{E}(x_7^2)$, $\mathrm{E}(x_3^2)$, and $\mathrm{E}(x_6^2)$ in $\mathrm{E}(x^2)$ can be found as follows:

$$\mathrm{E}(x_7^2) = [\mathrm{E}(x_7)]^2 + \mathrm{Var}(x_7)$$
$$= [1/\lambda_1 + 1/\lambda_2 + 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)]^2$$
$$+ \{1/\lambda_1^2 + 1/\lambda_2^2 + [2/\lambda_1^2 + 2/\lambda_3^2 - 2/(\lambda_1 + \lambda_3)^2]$$
$$- [1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)]^2\},$$

$$\mathrm{E}(x_3^2) = [\mathrm{E}(x_3)]^2 + \mathrm{Var}(x_3)$$
$$= [1/\lambda_2 + 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)]^2$$
$$+ \{1/\lambda_2^2 + [2/\lambda_1^2 + 2/\lambda_3^2 - 2/(\lambda_1 + \lambda_3)^2]$$
$$- [1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)]^2\},$$

$$\mathrm{E}(x_6^2) = 2/\lambda_2^2.$$

For SL(asyn), five states are identified in Table 6 (see also Figure 6). Steady-state distribution is obtained by solving:

Table 6 SL(asyn) States

States	Station 1	Station 2	Station 3
1	W	E	E
2	E	W	E
3	W	E	W
4	E	W	W
5	E	B	W

$$p_1 = p_3 \cdot \frac{\lambda_3}{\lambda_1 + \lambda_3},$$

$$p_2 = p_1 + p_4 \frac{\lambda_3}{\lambda_2 + \lambda_3},$$

$$p_3 = p_2 + p_5,$$

$$p_4 = p_3 \cdot \frac{\lambda_1}{\lambda_1 + \lambda_3},$$

$$p_1 + p_2 + p_3 + p_4 + p_5 = 1.$$

States 1, 2, and 3 are three possible system states after an output instant. The three associated output intervals are: $x_1 = S_1 + S_2 + S_3$ if system is in state 1; $x_2 = S_2 + S_3$ if system is in state 2; and $x_5 = S_3$ if system is in state 3.

$$CT_{\mathrm{SL(asyn)}} = \mathrm{E}(x) = (\mathrm{E}(x_1) \cdot p_1 + \mathrm{E}(x_2) \cdot (p_2 - p_1) + \mathrm{E}(x_5) \cdot p_5)/(p_5 + p_2);$$

$$\mathrm{E}(x^2) = (\mathrm{E}(x_1^2) \cdot p_1 + \mathrm{E}(x_2^2) \cdot (p_2 - p_1) + \mathrm{E}(x_5^2) \cdot p_5)/(p_5 + p_2);$$

$$\mathrm{V}_{\mathrm{SL(asyn)}} = \mathrm{E}(x^2) - [CT_{\mathrm{SL(asyn)}}]^2.$$

UL(I) has five Markovian states given in Table 7 (see also Figure 7). The transitional equations for state probabilities p_i $(i = 1, \ldots, 5)$ are:

$$p_1 = p_4 + p_5,$$

$$p_2 = \frac{\lambda_3}{\lambda_2 + \lambda_3} \cdot p_1,$$

$$p_3 = \frac{\lambda_2}{\lambda_2 + \lambda_3} \cdot p_1,$$

$$p_4 = \frac{\lambda_1}{\lambda_1 + \lambda_3} \cdot p_3,$$

$$p_1 + p_2 + p_3 + p_4 + p_5 = 1.$$

The immediate states after an output instant are states 3 and 5. The output intervals are: $x_3 = \max(t_1, t_3) + t_2$ if system is in state 3, and $x_5 = t_1 + t_2$ if system is in state 5,

$$CT_{\mathrm{UL(I)}} = \mathrm{E}(x) = (\mathrm{E}(x_3) \cdot p_3 + \mathrm{E}(x_5) \cdot p_2)/(p_2 + p_3).$$

Note that a portion of transitions into state 5 is from state 3 and has already been included into calculation by p_3, so it should be excluded from p_5. The leftover of p_5 after the exclusion is equivalent

Figure 6 State Transition Diagram for SL(asyn)

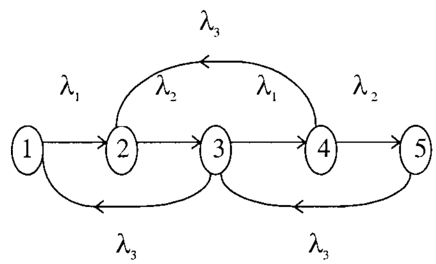
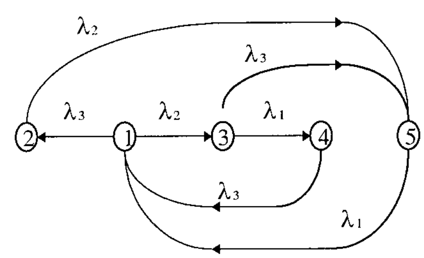


Table 7 UL(I) States

State	Station 1	Station 2	Station 3
1	E	W	W
2	E	W	B
3	W	E	W
4	B	E	W
5	W	E	B

Figure 7 State Transition Diagram for UL(I)



to p_2. That is why p_2 is used in the expression for $\mathrm{E}(x)$. The $V_{\mathrm{UL(I)}}$ is derived as follows:

$$\mathrm{E}(x_3) = 1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3) + 1/\lambda_2,$$

$$\mathrm{E}(x_5) = 1/\lambda_1 + 1/\lambda_2,$$

$$\mathrm{E}(x_3^2) = [\mathrm{E}(x_3)]^2 + \mathrm{Var}(x_3),$$

$$\mathrm{Var}(x_3) = [2/\lambda_1^2 + 2/\lambda_3^2 - 2/(\lambda_1 + \lambda_3)^2] - [1/\lambda_1 + 1/\lambda_3 - 1/(\lambda_1 + \lambda_3)]^2 + 1/\lambda_2^2,$$

$$\mathrm{E}(x_5^2) = 1/\lambda_1^2 + 1/\lambda_2^2 + [\mathrm{E}(x_5)]^2,$$

$$\mathrm{E}(x^2) = (\mathrm{E}(x_3^2) \cdot p_3 + \mathrm{E}(x_5^2) \cdot p_2)/(p_2 + p_3),$$

$$V_{\mathrm{UL(I)}} = \mathrm{Var}(x) = \mathrm{E}(x^2) - [\mathrm{E}(x)]^2.$$

Using the similar approach for UL(O) as shown above, we obtain the following results:

$$CT_{\mathrm{UL(I)}} = CT_{\mathrm{SL(asyn)}} = \frac{(\lambda_2^{2}*\lambda_1*\lambda_3 + \lambda_2^{2}*\lambda_1^2 + \lambda_3^{2}*\lambda_1^2 + \lambda_3^{3}*\lambda_1 + \lambda_2*\lambda_1^{2}*\lambda_3 + 2*\lambda_3^{2}*\lambda_1*\lambda_2 + \lambda_3^{2}*\lambda_2^2 + \lambda_3^{3}*\lambda_2}{(\lambda_1 + \lambda_3)*\lambda_1*\lambda_3*(\lambda_2 + \lambda_3)*\lambda_2},$$

$$CT_{\mathrm{UL(O)}} = \frac{3*\lambda_3^{2}*\lambda_1*\lambda_2 + 2*\lambda_3^{3}*\lambda_2 + 2*\lambda_3^{2}*\lambda_1^2 + 2*\lambda_3^{3}*\lambda_1 + 2*\lambda_2*\lambda_1^{2}*\lambda_3 + \lambda_2^{2}*\lambda_1*\lambda_3 + \lambda_3^{2}*\lambda_2^2 + \lambda_2^{2}*\lambda_1^2}{(2*\lambda_3 + \lambda_2)*\lambda_1*\lambda_2*\lambda_3*(\lambda_1 + \lambda_3)},$$

$$\begin{aligned} V_{\mathrm{SL(asyn)}} = {} & (2\lambda_3^3\lambda_1\lambda_2^4 + 2\lambda_3^3\lambda_1^4\lambda_2 + 4\lambda_2^2\lambda_3^3\lambda_1^3 + 2\lambda_2^2\lambda_3^5\lambda_1 + 2\lambda_2\lambda_3^5\lambda_1^2 + 2\lambda_3\lambda_1^4\lambda_2^3 + 4\lambda_1^2\lambda_3^4\lambda_2^2 + 4\lambda_3^2\lambda_1^3\lambda_2^3 + 4\lambda_3^3\lambda_1^2\lambda_2^3 + 2\lambda_3\lambda_1^3\lambda_2^4 + 4\lambda_1\lambda_2^3\lambda_3^4 \\ & + \lambda_3^2\lambda_1^4\lambda_2^2 + 4\lambda_1^3\lambda_3^4\lambda_2 + \lambda_1^4\lambda_3^4 + \lambda_3^4\lambda_2^4 + \lambda_1^4\lambda_2^4 + 2\lambda_3^5\lambda_2^3 + \lambda_3^6\lambda_2^2 + \lambda_3^6\lambda_1^2 + 2\lambda_3^5\lambda_1^3 + \lambda_3^2\lambda_1^2\lambda_2^4) \\ & \div (\lambda_2^2\lambda_1^2(\lambda_2\lambda_1 + \lambda_2\lambda_3 + \lambda_3\lambda_1 + \lambda_3^2)^2\lambda_3^2), \end{aligned}$$

$$\begin{aligned} V_{\mathrm{UL(O)}} = {} & (-\lambda_2^4\lambda_3^2\lambda_1^2 + 2\lambda_2^4\lambda_3\lambda_1^3 + 4\lambda_3^3\lambda_1^4\lambda_2 + 8\lambda_3^4\lambda_1^3\lambda_2 + 4\lambda_3^5\lambda_1^2\lambda_2 + 4\lambda_1^4\lambda_2^3\lambda_3 + 16\lambda_2^2\lambda_3^5\lambda_1 + 2\lambda_2^4\lambda_3^3\lambda_1 + 12\lambda_2^3\lambda_3^4\lambda_1 + 10\lambda_2^2\lambda_3^3\lambda_1^3 \\ & + 9\lambda_2^2\lambda_3^4\lambda_1^2 + 2\lambda_2^3\lambda_3^3\lambda_1^2 + 8\lambda_2^3\lambda_3^2\lambda_1^3 + 4\lambda_3^2\lambda_1^4\lambda_2^2 + 8\lambda_3^6\lambda_2^2 + 6\lambda_3^5\lambda_2^3 + 4\lambda_3^4\lambda_1^4 + 8\lambda_3^5\lambda_1^3 + 4\lambda_3^6\lambda_1^2 + \lambda_3^4\lambda_2^4 + \lambda_1^4\lambda_2^4) \\ & \div ((2\lambda_3 + \lambda_2)^2\lambda_2^2(\lambda_1 + \lambda_3)^2\lambda_3^2\lambda_1^2), \end{aligned}$$

$$CT_{\mathrm{UL(O)}} - CT_{\mathrm{UL(I)}} = \frac{\lambda_3\lambda_1}{(\lambda_3 + \lambda_2)(\lambda_3 + \lambda_1)(2\lambda_3 + \lambda_2)} > 0, \tag{10}$$

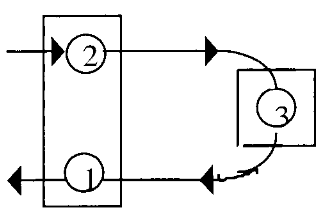
$$V_{\mathrm{SL(asyn)}} - V_{\mathrm{UL(I)}} = \frac{2(\lambda_3 + \lambda_2 + \lambda_1)}{(\lambda_1 + \lambda_3)^2(\lambda_2 + \lambda_3)} > 0, \tag{11}$$

$$\begin{aligned} V_{\mathrm{UL(O)}} - V_{\mathrm{UL(I)}} = {} & (4\lambda_2^3\lambda_3^2\lambda_1 + 4\lambda_3^4\lambda_1^2 + 14\lambda_3^3\lambda_1^2\lambda_2 + 2\lambda_2^3\lambda_3^3 + 4\lambda_2^3\lambda_1^2\lambda_3 + 14\lambda_3^2\lambda_2^2\lambda_1^2 + 2\lambda_2^3\lambda_1^3 + 10\lambda_3^5\lambda_2 + 16\lambda_2^2\lambda_3^3\lambda_1 + 4\lambda_3^6 + 10\lambda_2^2\lambda_1^3\lambda_3 \\ & + 8\lambda_3^5\lambda_1 + 20\lambda_2\lambda_3^4\lambda_1 + 8\lambda_3^3\lambda_1^3 + 16\lambda_3^2\lambda_1^3\lambda_2 + 8\lambda_2^2\lambda_3^4 + 3\lambda_3^2\lambda_1^4 + 2\lambda_3\lambda_1^4\lambda_2)/((2\lambda_3 + \lambda_2)^2(\lambda_1 + \lambda_3)^2\lambda_1^2(\lambda_2 + \lambda_3)^2) > 0, \end{aligned} \tag{12}$$

$$\begin{aligned} V_{\mathrm{UL(O)}} - V_{\mathrm{SL(asyn)}} = {} & (4\lambda_3^6 + 10\lambda_3^5\lambda_2 + 8\lambda_3^5\lambda_1 + 20\lambda_2\lambda_3^4\lambda_1 + 8\lambda_2^2\lambda_3^4 - 4\lambda_3^4\lambda_1^2 + 2\lambda_2^3\lambda_3^3 - 10\lambda_3^3\lambda_1^2\lambda_2 + 16\lambda_2^2\lambda_3^3\lambda_1 + 4\lambda_2^3\lambda_3^2\lambda_1 + 3\lambda_3^2\lambda_1^4 \\ & - 12\lambda_3^2\lambda_2^2\lambda_1^2 - 8\lambda_2^3\lambda_1^2\lambda_3 + 2\lambda_3\lambda_1^4\lambda_2 - 2\lambda_1^2\lambda_2^4)/((\lambda_2\lambda_1 + \lambda_2\lambda_3 + \lambda_3\lambda_1 + \lambda_3^2)^2\lambda_1^2(2\lambda_3 + \lambda_2)^2) \end{aligned} \tag{13}$$

(sign contingent on values of λ_1, λ_2, λ_3).

Figure 8 Reversed UL (RUL)



Appendix 2.

Reversibility. A production line is reversible if the mean production rate remains invariant under reversal of the production line (Muth 1979). Line reversal means every unit passes through the stations in reverse order, that is, beginning with the last station and ending with the first station. Muth showed that a straight line is reversible for any arbitrarily distributed task times. We find that when the line has a U-shape layout, the reversibility property of the U-line depends on the work rule also (in addition to properties of task times). Consider the 3-task, 2-worker example of Figure 2 (n = 3); the reversed UL (RUL) is shown in Figure 8. When RUL uses the input-first work rule (denoted as RUL(I)), its average production rate is $1/[E(\max(S_1 + S_2, S_3))]$, the same as for UL(I). So, for the 3-task, 2-worker example, UL(I) is reversible if RUL also uses the input-first work rule. This result can be easily extended to UL with n stations. With the output-first rule, UL(O) has two types of cycles as shown in Figure 9.

Figure 9 Production Cycles for UL(O)

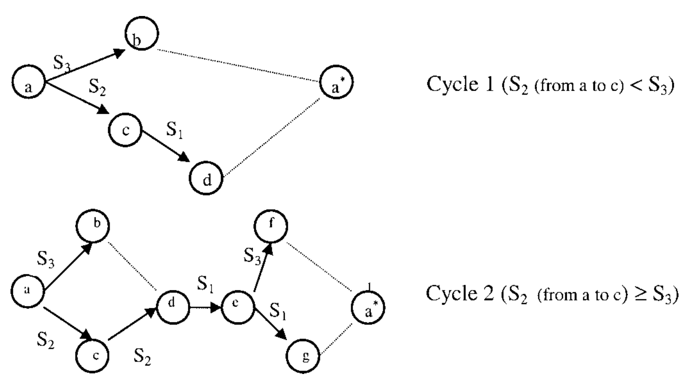


Figure 10 **UL Layout for $n = 3$**

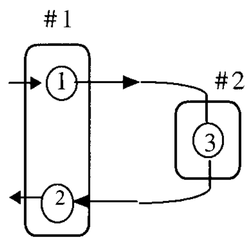
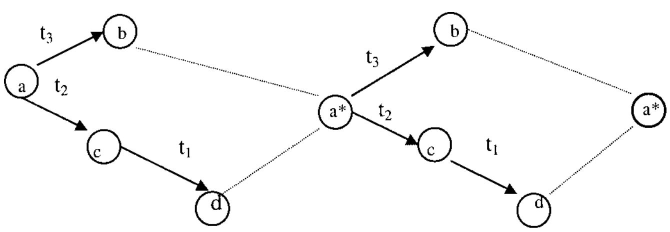


Figure 11 **Production Cycles for UL(I)**



The expected lengths of cycle 1 and cycle 2 are:

$$\mathrm{E(cycle\ 1)} = \mathrm{E}(\max(S_2 + S_1, S_3) \mid S_2 < S_3),$$

$$\mathrm{E(cycle\ 2)} = \mathrm{E}((S_1 + S_2 \mid S_2 \geq S_3) + \max(S_1 + S_2, S_3 + S_2)).$$

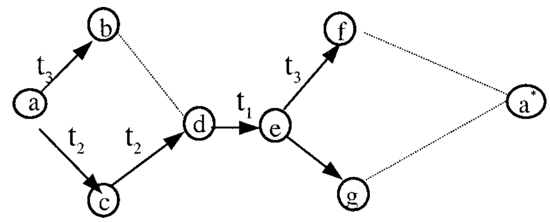
The average cycle length can be calculated as follows:

$$\text{Average cycle time} = \frac{\mathrm{E(cycle\ 1)} \cdot \mathrm{P}(S_2 < S_3) + \mathrm{E(cycle\ 2)} \cdot \mathrm{P}(S_3 \leq S_2)}{\mathrm{P}(S_2 < S_3) + 2\mathrm{P}(S_3 \leq S_2)}.$$

Because two units are produced in every ''cycle 2'' while only one unit is produced in every ''cycle 1,'' the weight of 2 is given to $\mathrm{P}(S_2 \geq S_3)$ in the denominator. Now consider RUL under the output-first rule (RUL(O)); the average cycle time for RUL(O) can be obtained simply by interchanging S_1 with S_2. In general, S_1 has different distribution than S_2. Consequently, average cycle times are different for RUL(O) and UL(O), and so are the mean production rates. This small example shows that the reversibility property of UL depends on the selection of the work rule, which is not an issue in SL.

Proof of Result 1, Part (c). We prove the result for $n = 3$. For $n = 3$, the UL layout becomes that shown in Figure 10. The production cycles for UL(I) can be represented by the network diagram in Figure 11. The initial node (node a in Figure 11) represents the instant when stations 2 and 3 begin working on the units at these stations. After t_2 time periods, a finished unit will emerge from station 2 and we reach node c in Figure 11. With input-first rule, worker 1 will go back to station 1 after finishing the task at station 2. At station 1, the worker will pull a unit of raw material and start working on it. After both stations 1 and 3 have finished their tasks, worker 1 will pull the unit from station 3 to station 2 while worker 2 will pull the unit from station 1 to station 3, and we reach node

Figure 12 **Production Cycles for UL(O) with $t_2 \geq t_3$**



a^*. This brings us back to the state when both stations 2 and 3 begin working on units on these stations, and the next cycle begins. The dotted lines show that node a^* is reached when both stations 1 and 3 have finished their tasks. Note that if station 3 has finished its task and worker 1 is busy at station 1, the unit will have to wait at station 3 because worker 1 is not free to pull it to station 2. The length of the production cycle is equal to $\max(t_1 + t_2, t_3)$. The proof can be easily extended for $n > 3$. □

Proof of Result 2. For the case when $t_2 < t_3$, it is possible to see that worker 1 will always work at station 1 after completing the task at station 2 as in UL(I). UL(O) is the same as UL(I) in this case, and a proof follows from (c) in Result 1.

For the case when $t_2 \geq t_3$, the production cycles are as in Figure 12. Because $t_3 \leq t_2$, task 3 is done before task 2 finishes. Upon completion of task 2, with output-first rule, the worker will pull the unit from station 3 to station 2 and spend t_2 time periods on the unit; its completion is represented by node d. (The production line is empty at node d.) Then, the worker will go to station 1 and spend t_1 time periods on a unit. After task 1 is done, worker 2 will pull the unit to station 3 while worker 1 will pull a unit to station 1. The next cycle will start after both tasks 1 and 3 are done. It follows that 2 units are produced every $2 \cdot t_2 + t_1 + \max(t_1, t_3)$ time periods, which equals $2 \max(t_1 + t_2, t_3) + \max(0, t_3 - t_1)$ because $t_2 \geq t_3$. □

Proof of Result 3. See Baker et al. (1990) for a proof of (a) and (b). The proof for (c) in Result 1 can be easily extended for (c) here. A proof similar to that of Result 2 can be provided for (d); the production cycles along with lengths are shown in the discussion of reversibility. □

Proof of Result 4. We first prove (c) and then (a) and (b). A proof for (d) is provided in Appendix 1. □

Proof of (c). Consider the two consecutive production cycles for SL(asyn) and UL(I) given in Figure 13; the output points are also shown. Let S_i^l denote the task time of station i in the second production cycle. Note that S_i and S_i^l are identically and independently distributed random variables. So, a realization of S_i^l is independent of a realization of S_i.

It is possible to see that $Y_{\mathrm{SL(asyn)}} = \max(0, S_1 + S_2 - S_3) + S_3^l$. Then we get

Figure 13 Production Cycles for SL(asyn) and UL(I)

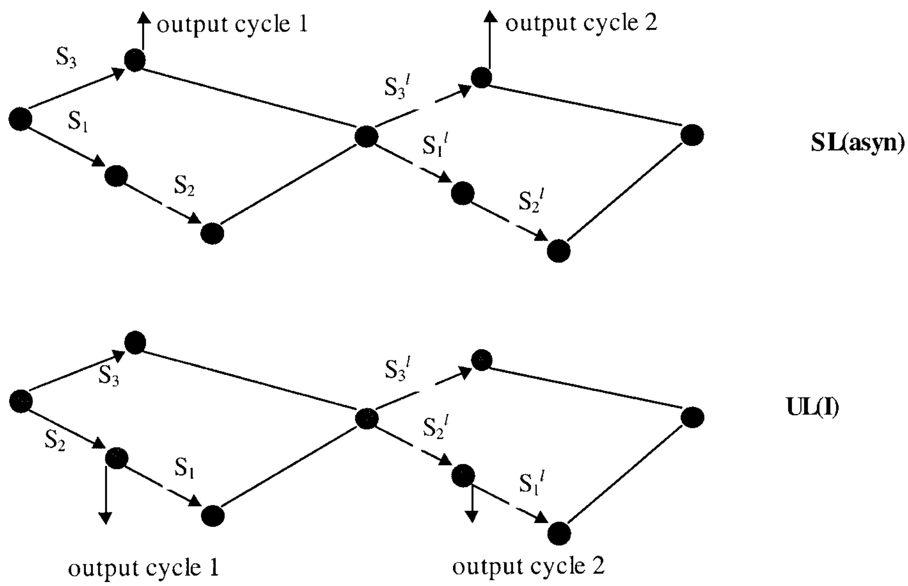
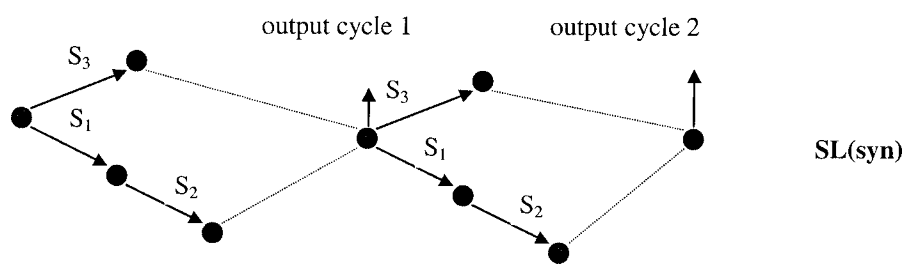


Figure 14 Output Points for SL(syn)



$$Y_{\mathrm{SL(asyn)}} \cong \begin{cases} S_3^l & \text{when } S_3 \text{ is large compared to } S_1 + S_2, \\ S_1 + S_2 - S_3 + S_3^l & \\ \quad \text{when } S_3 \text{ is small,} \end{cases}$$

$$V_{\mathrm{SL(asyn)}} \cong \begin{cases} v(S_3) & \text{when } S_3 \text{ is large compared to } S_1 + S_2, \\ v(S_1) + v(S_2) + 2v(S_3) & \\ \quad \text{when } S_3 \text{ is small.} \end{cases}$$

By S_3 large compared to $S_1 + S_2$, we mean we know a priori that the random variable S_3 is surely larger than or equal to the random variable $S_1 + S_2$.

We can also see that $Y_{\mathrm{UL(I)}} = S_1 + \max(0, S_3 - S_1 - S_2) + S_2^l$. We get

$$Y_{\mathrm{UL(I)}} \cong \begin{cases} S_3 - S_2 + S_2^l & \text{when } S_3 \text{ is large compared to } S_1 + S_2, \\ S_1 + S_2^l & \text{when } S_3 \text{ is small,} \end{cases}$$

$$V_{\mathrm{UL(I)}} \cong \begin{cases} v(S_3) + 2v(S_2) & \text{when } S_3 \text{ is large compared to } S_1 + S_2, \\ v(S_1) + v(S_2) & \text{when } S_3 \text{ is small.} \end{cases}$$

The above analysis shows that UL(I) has a higher variance than SL(asyn) if S_3 is large compared to $S_1 + S_2$, and a lower variance if S_3 is small compared to $S_1 + S_2$. □

PROOF OF (A). The output points for SL(syn) ($n = 3$) are as shown in Figure 14. It is easy to see that

$$Y_{\mathrm{SL(syn)}} = \max(S_1 + S_2, S_3) = \max(S_1, S_3 - S_2) + S_2.$$

From the proof of (c), we know

$$Y_{\mathrm{UL(I)}} = S_1 + \max(0, S_3 - S_1 - S_2) + S_2^I$$
$$= \max(S_1, S_3 - S_2) + S_2^I.$$

We can easily extend the above expressions for $n \geq 3$:

$$Y_{\mathrm{SL(syn)}} = \max(S_1 + S_2, S_3, \ldots, S_n)$$
$$= \max(S_1, S_3 - S_2, \ldots, S_n - S_2) + S_2,$$
$$Y_{\mathrm{UL(I)}} = S_1 + \max(0, S_3 - S_1 - S_2, \ldots, S_n - S_1 - S_2) + S_2^I$$
$$= \max(S_1, S_3 - S_2, \ldots, S_n - S_2) + S_2^I.$$

The proof follows from the observation that S_2 is negatively correlated[1] with $\max(S_1, S_3 - S_2, \ldots, S_n - S_2)$ while S_2^I is independent of $\max(S_1, S_3 - S_2, \ldots, S_n - S_2)$. □

We next provide a proof for (b).

Proof of (b). We have $Y_{\mathrm{SL(syn)}} = \max(S_1 + S_2, S_3)$ and $Y_{\mathrm{SL(asyn)}} = \max(0, S_1 + S_2 - S_3) + S_3^I$. We can write $Y_{\mathrm{SL(syn)}} = \max(0, S_1 + S_2 - S_3) + S_3$. The proof follows from the observation that S_3 is negatively correlated with $\max(0, S_1 + S_2 - S_3)$, while S_3^I is independent of $\max(0, S_1 + S_2 - S_3)$. □

References

Baker, K. R., S. G. Powell, D. F. Pyke. 1990. Buffered and unbuffered assembly systems with variable processing times. *J. Manufacturing Oper. Management.* **3** (3) 200–223.

Bartholdi, J. J., L. A. Bunimovich, D. Eisenstein. 1999. Dynamics of 2- and 3-worker "bucket brigade" production lines. *Oper. Res.* **47** (3) 481–491.

——, D. Eisenstein. 1996. A production line that balances itself. *Oper. Res.* **44** (1) 21–34.

Baybars, Ilker. 1986. A survey of exact algorithms for the simple assembly line. *Management Sci.* **32** (8) 909–932.

Bischak, D. P. 1996. Performance of a manufacturing module. *IIE Trans.* **28** (28) 723–733.

Carraway, R. L. 1989. A dynamic programming approach to stochastic assembly line balancing. *Management Sci.* **35** (4) 459–467.

Harrison, M. J., L. Wein. 1990. Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Oper. Res.* **38** (6) 1052–1064.

Hopp, W. J., M. L. Spearman. 1996. *Factory Physics.* Irwin McGraw-Hill, New York, NY.

Kao, E. P. C. 1976. A preference order dynamic programming method for assembly line balancing. *Management Sci.* **22** (10) 1097–1104.

Kumar, P. R. 1993. Re-entrant lines. *Queuing Systems.* **13** 87–110.

Miltenburg, G. J., J. Wijngaard. 1994. The u-line balancing problem. *Management Sci.* **40** (10) 1378–1388.

Muth, E. J. 1979. The reversibility property of production lines. *Oper. Res.* **25** (2) 152–158.

Talbot, B., J. Patterson, W. Gehrlein. 1986. A comparative evaluation of heuristic line balancing techniques. *Management Sci.* **32** (4) 430–454.

Urban, T. L. 1998. Optimal balancing of u-shaped assembly lines. *Management Sci.* **44** (5) 738–741.

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

[1] We thank Professor Jayshankar Swaminathan of the University of North Carolina for this observation.

The consulting Senior Editor for this manuscript was L. Joseph Thomas. This manuscript was received on January 27, 1998, and was with the authors for 986 days for 6 revisions. The average review cycle time was 36.17 days.