

ANALYSIS AND OPTIMIZATION OF A U-SHAPED PRODUCTION LINE

Katsuhisa Ohno Koichi Nakade
Nagoya Institute of Technology

(Received August 21, 1995; Revised January 16, 1996)

Abstract In the *just-in-time* context, parts are often processed by a *single-unit production and conveyance system* (called "*ikko-nagashi*" in Japanese) without conveyors. The U-shaped layout, in which each multi-function worker takes charge of several machines, has been introduced as an implementation of this concept. Presently the layout is gaining an increasing popularity due to the low running cost.

In this paper, first we deal with the U-shaped production line with a single multi-function worker. We derive his waiting time and a cycle time of the line when processing times of items, operation times, and his walking times between machines are constants. Then we deal with a U-shaped production line with multiple workers. We derive the overall cycle time of this line, and consider an optimal worker allocation problem that minimizes the overall cycle time when the number of workers is given. In particular, it is shown that the U-shaped layout is superior to the linear layout for lines with one or two workers. We also discuss the case where those processing, operation and walking times are stochastic.

1. Introduction

In a conveyor system for mass production as in the Ford system, each station processes just one item in one cycle time, where the cycle time is the time-interval between two successive outputs. The sums of necessary operation and processing times are intended to be equal among the stations, the items are processed synchronously among the stations, and there exist no items between adjacent stations.

In the *just-in-time* (JIT) production system, the above concept, which is called a *single-unit production and conveyance* ("*ikko-nagashi*," in Japanese), is applied to a production line without conveyors which manufactures different kinds of relatively small parts (Monden [3], p.107). To achieve this at a low production cost, a U-shaped layout is used with multi-function workers. The U-shaped production line with three workers and ten machines is shown in Figure 1. When the entrance and exit of items are near as shown in Figure 1, we call this layout a U-shaped layout, and if the same worker handles both machines at the entrance and exit in the U-shaped layout then we call this layout a U-shaped production line.

The multi-function worker takes charge of multiple machines, and visits each of them once in one cycle. When he arrives at one of these machines, he waits for the end of processing of the preceding item if it is not completed, and then operates the items and walks to the next machine. The operation consists of detaching the processed item from the machine, putting it on a chute to roll in front of the next machine, attaching the new item to the machine, and switching it on. The cycle time of the worker is the time-interval between his consecutive arrivals at his first machine, and consists of the waiting times for the end of processing, operation times and walking times between machines.

In the JIT production system, two kinds of Kanbans, that is, a production ordering and a withdrawal Kanbans are used as tools to control production and withdrawal quantities in

each production line. In the U-shaped line, the same worker inputs a new item and outputs a completed product. Consequently, he can observe changes of two kinds of Kanbans and respond to them promptly. Since a new item enters the system only after one completed product exits, the work-in-process in the system is always constant. Further, there exist more possible allocations of the workers to machines than in the linear layout. Therefore, when the demand changes we can more appropriately reallocate the workers to machines so that the cycle times of workers are balanced. That is, the U-shaped layout can be more properly adapted to the changes of the circumstances than the linear layout.

In this paper, we first consider the U-shaped production line with just one multi-function worker. We analyze his waiting time and the cycle time. Then we consider the overall cycle time of the U-shaped line with more than one multi-function worker, which is the maximum of the cycle times of all workers. It is noted that its reciprocal gives the throughput, or the production rate of finished products. Moreover, we consider an optimal worker allocation problem that minimizes the overall cycle time.

In Section 2, we explain the U-shaped production line with a single worker, and analyze his waiting time and the cycle time of this line, when the operation, walking and processing times are constants. We show that the n -th cycle time becomes constant for $n \geq 2$, and that after several cycles the worker waits for the completion of processing of at most one specified machine.

Recently, Miltenberg and Wijngaard [2] considered the line balancing problem of the U-shaped line with constant operation times, no waiting times and no walking times. They discussed the optimal machine allocation problem to workers (which they called stations) under the constraints on the orders of machines in which the items are processed, like the assembly line balancing problems (Baybars [1], for example). In the U-shaped line, however, the walking times should be taken into account to derive the exact cycle time. In addition, it is possible for the worker to wait for the end of processing at a machine for an allocation, because the time interval from departure to next arrival of the worker at the machine may exceed the processing time at the machine. Therefore, the problem which they discussed does not represent the real features of the U-shaped line. In Section 3, we consider a production line with I workers and K machines, and derive the overall cycle time of this line under a given allocation of workers to machines. Then we discuss the optimal worker allocation problem that minimizes the overall cycle time of this line. It is shown that the problem can be formulated into a combinatorial optimization problem. We examine the optimal worker allocation problem with one or two workers in a production line with K machines placed at the same distance. This will reveal advantages of the U-shaped layout over the linear layout.

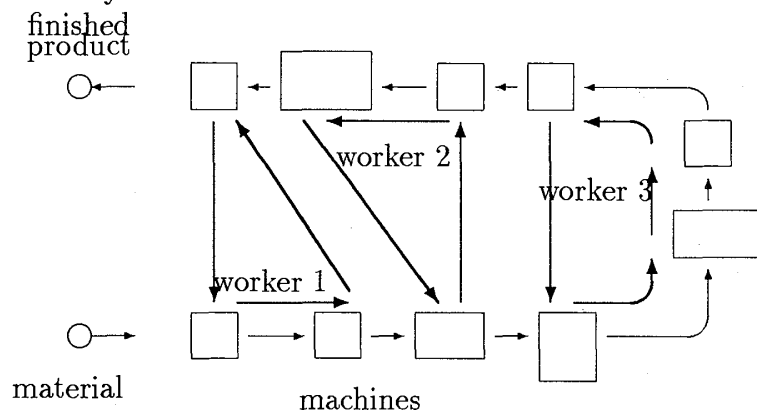


Figure 1. A U-shaped Production Line

We can further reduce an overall cycle time by admitting what Toyota calls mutual relief movement ([3], p.114). This means that a worker who has finished his own operations in one cycle helps another adjacent worker. This, however, is not taken into account in this paper, because the problem becomes more complicated.

If multiple kinds of items are processed in this line, the processing times and operation times are not constant. In addition, the operation and walking times of the worker may fluctuate because of his weariness and learning effect. In Section 4, we deal with the case where the processes of operation, walking and processing times are stochastic. In particular, we discuss the case where the sequences of random variables in these processes are independent and identically distributed and there is a bottleneck machine such that the sum of processing and operation times of this machine is larger than that of any other machine with probability one. It can be shown that the worker waits for the completion of processing at the bottleneck machine in all cycles.

2. Cycle and Waiting Times of a Multi-Function Worker

In this section we consider the U-shaped production line with a single multi-function worker, which is shown in Figure 2. The worker handles machines 1 through K . The facility has enough raw material in front of machine 1. The material is processed at machines 1, 2, \dots , K , sequentially, and departs from the system as a finished product. Let $\hat{K} = \{1, \dots, K\}$. When the worker arrives at machine $k \in \hat{K}$, if the processing of the preceding item is completed, then he removes it from machine k , sends it to machine $k + 1$, attaches the present item to machine k and switches it on. After the operation at machine k , he walks to machine $k + 1$. If the preceding item is still in process at his arrival, then he waits for the end of the processing before the operation.

It is assumed as an initial condition that at time 0, there is one item on each machine, which has been already processed at this machine. That is, in the first cycle the worker operates without waiting at all machines. In this and next sections, we assume that the processing, operation and walking times are constants at each machine. This assumption is satisfied when one kind of products are produced and the worker is well experienced in the operation. We use the following notations: for $k \in \hat{K}$ and $n \in \mathcal{Z}^+ = \{1, 2, \dots\}$,

i_k : the processing time at machine k ,

s_k : the operation time of the worker at machine k ,

r_k : the walking time from machine k to machine $k + 1$ (K to 1, if $k = K$),

$W_k(n)$: the waiting time of the worker at machine k in the n -th cycle,

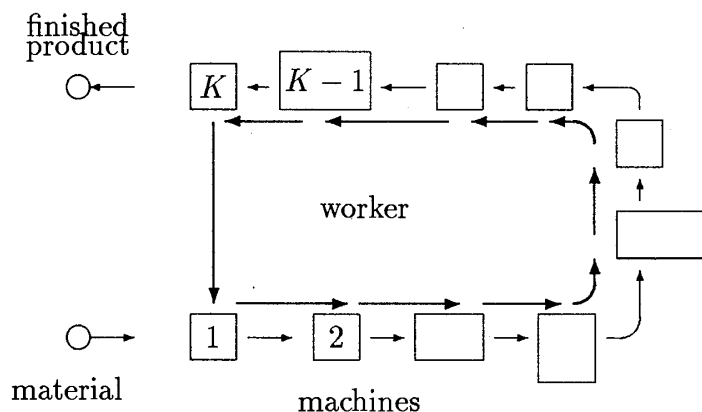


Figure 2. A U-shaped Production Line with a Single Worker

$C(n)$: the n -th cycle time,
 $a \vee b = \max\{a, b\}$, $a \wedge b = \min\{a, b\}$, $[a]^+ = \max\{0, a\}$.

Figure 3 illustrates the behavior of the worker and the above defined variables. The initial condition implies that

$$W_k(1) = 0 \quad \text{for } k \in \hat{K}, \quad \text{and} \quad C(1) = \sum_{k \in \hat{K}} (s_k + r_k). \tag{1}$$

Consider the n -th cycle time for $n \geq 2$. If the worker does not wait at any machine then the cycle time is simply the sum of all operation and walking times. Since one item is processed and operated at each machine in one cycle, the cycle time must be greater than or equal to the maximum of the sums of the processing and operation times among all the machines. If the worker starts from the machine with the maximum sum, then the cycle time will be equal to the maximum of the maximum sum and the sum of all operation and walking times. That is, the cycle time will be given by

$$\max_{k \in \hat{K}} \{i_k + s_k\} \vee \sum_{k \in \hat{K}} (s_k + r_k), \tag{2}$$

and then the total waiting time of the worker in one cycle is given by

$$\left\{ \max_{k \in \hat{K}} \{i_k + s_k\} \vee \sum_{k \in \hat{K}} (s_k + r_k) \right\} - \sum_{k \in \hat{K}} (s_k + r_k) = \max_{k \in \hat{K}} [y_k]^+ \tag{3}$$

where for $k \in \hat{K}$,

$$y_k = i_k + s_k - \sum_{j \in \hat{K}} (s_j + r_j). \tag{4}$$

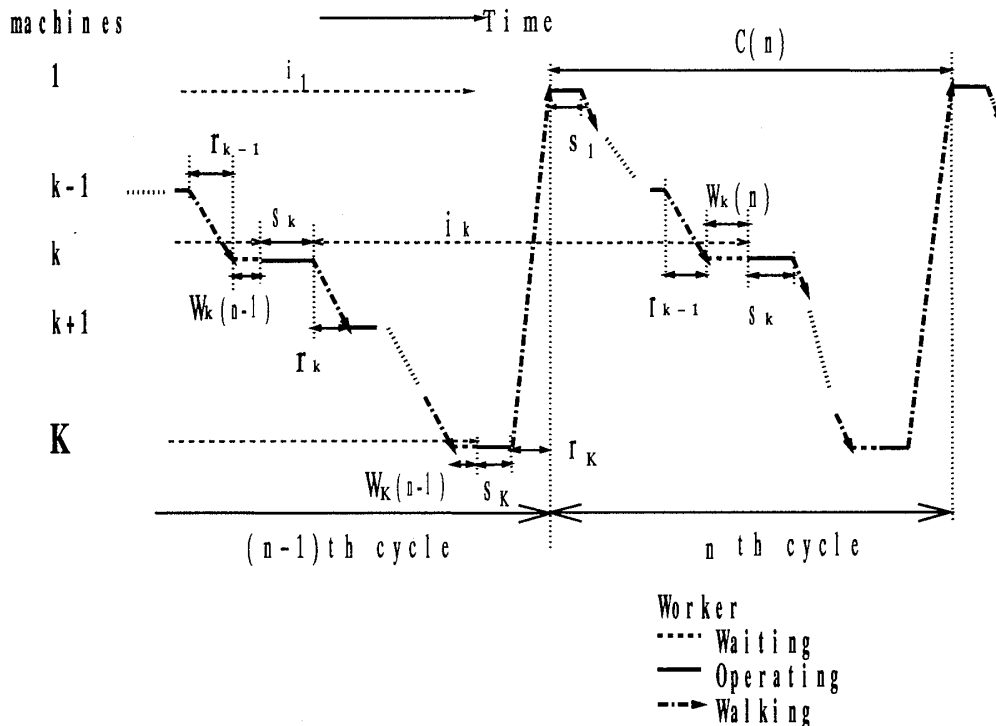


Figure 3. Behavior of a Multi-Function Worker

In the following, we derive the n -th waiting time at machine k for all $k \in \hat{K}$ and $n \in \mathcal{Z}^+$, and show that (2) and (3) hold in the n -th cycle for all $n \geq 2$. In the $(n-1)$ -th cycle where $n \geq 2$, when the worker finishes the operation at machine k and starts walking to machine $k+1$, machine k begins the processing of the $(n-1)$ -th item. When he returns to machine k in the n -th cycle, if the machine completes the $(n-1)$ -th processing, then he begins the operation for the n -th item without waiting. If the $(n-1)$ -th item is still in process at his return, then he waits for its completion. Figure 3 shows that the time from the $(n-1)$ -th departure to the n -th arrival of the worker at machine k is given by

$$\begin{aligned} r_k + \sum_{j>k} (W_j(n-1) + s_j + r_j) + \sum_{j<k} (W_j(n) + s_j + r_j) \\ = \sum_{j>k} W_j(n-1) + \sum_{j<k} W_j(n) + \sum_{j \neq k} s_j + \sum_{j \in \hat{K}} r_j. \end{aligned}$$

Since the waiting time at machine k is the time difference between the processing time and the interarrival time, it holds that for $n \geq 2$,

$$\begin{aligned} W_k(n) &= \left[i_k - \left\{ \sum_{j>k} W_j(n-1) + \sum_{j<k} W_j(n) + \sum_{j \neq k} s_j + \sum_{j \in \hat{K}} r_j \right\} \right]^+ \\ &= \left[y_k - \sum_{j>k} W_j(n-1) - \sum_{j<k} W_j(n) \right]^+, \quad \text{for } k \in \hat{K}, \end{aligned} \quad (5)$$

$$\text{and } C(n) = \sum_{k \in \hat{K}} (W_k(n) + s_k + r_k). \quad (6)$$

Let k^* be the smallest index of machines with the maximum value of y_k , $k \in \hat{K}$. That is,

$$k^* = \min\{k; y_k = \max_{j \in \hat{K}} y_j\}.$$

Define

$$y_{m(k)} = \max_{j \leq k} y_j, \quad y_{m(0)} = 0,$$

$$\text{and } A_k(n) = \sum_{j=1}^k W_j(n), \quad \text{for } k \in \hat{K} \text{ and } n \in \mathcal{Z}^+.$$

If $y_{k^*} \leq 0$, then by (5) $W_k(n) = 0$ for all $k \in \hat{K}$ and $n \in \mathcal{Z}^+$, and hence $A_k(n) = 0$ for all $k \in \hat{K}$ and $n \in \mathcal{Z}^+$.

We assume temporarily that $y_{k^*} > 0$. We show that for $n \geq 2$ and all $k \in \hat{K}$,

$$A_k(n) = \left[(n-1)y_{m(k)} - (n-2)y_{k^*} \right]^+. \quad (7)$$

Let $n = 2$. For $k = 1$, by (1) and (5) $A_1(2) = W_1(2) = [y_1]^+$. Suppose that $A_k(2) = [y_{m(k)}]^+$. Then (1) and (5) lead to

$$\begin{aligned} A_{k+1}(2) &= W_{k+1}(2) + A_k(2) = [y_{k+1} - A_k(2)]^+ + A_k(2) \\ &= [y_{k+1}] \vee A_k(2) = [y_{m(k+1)}]^+. \end{aligned}$$

Hence

$$A_k(2) = [y_{m(k)}]^+, \quad \text{for all } k \in \hat{K}.$$

We suppose that (7) holds for n , and show that (7) also holds for $n + 1$. Since $A_K(n) = y_{k^*}$ from the assumption $y_{k^*} > 0$, (5) and the inductive assumption lead to

$$\begin{aligned} A_1(n+1) &= [y_1 + A_1(n) - A_K(n)]^+ \\ &= [y_1 + [(n-1)y_1 - (n-2)y_{k^*}]^+ - y_{k^*}]^+ \\ &= [\{ny_1 - (n-1)y_{k^*}\} \vee (y_1 - y_{k^*})]^+ \\ &= [ny_1 - (n-1)y_{k^*}]^+, \end{aligned}$$

where the last equality holds because $y_1 - y_{k^*} \leq 0$. Then by (5),

$$\begin{aligned} A_2(n+1) &= W_2(n+1) + A_1(n+1) \\ &= [y_2 - A_1(n+1) - (A_K(n) - A_2(n))]^+ + A_1(n+1) \\ &= \max\{y_2 - A_K(n) + A_2(n), A_1(n+1)\} \\ &= \max\{y_2 - y_{k^*} + [(n-1)y_{m(2)} - (n-2)y_{k^*}]^+, [ny_{m(1)} - (n-1)y_{k^*}]^+\} \\ &= \max\{y_2 + (n-1)y_{m(2)} - (n-1)y_{k^*}, y_2 - y_{k^*}, [ny_{m(1)} - (n-1)y_{k^*}]^+\} \\ &= [\max\{y_2 + (n-1)y_{m(2)} - (n-1)y_{k^*}, ny_{m(1)} - (n-1)y_{k^*}\}]^+ \\ &= [ny_{m(2)} - (n-1)y_{k^*}]^+, \end{aligned}$$

where the last equality holds because

$$ny_{m(k)} = \max\{y_k + (n-1)y_{m(k)}, ny_{m(k-1)}\} \quad \text{for all } k \in \{2, \dots, K\}.$$

Similarly, it holds that for all $k \in \{3, \dots, K\}$

$$A_k(n+1) = [ny_{m(k)} - (n-1)y_{k^*}]^+.$$

Therefore (7) holds for all $k \in \hat{K}$ and all $n \geq 2$. Note that (7) also holds when $y_{k^*} \leq 0$.

In particular, $m(k) = k^*$ for $k \geq k^*$, and hence for $n \geq 2$

$$A_{k^*}(n) = A_{k^*+1}(n) = \dots = A_K(n) = [y_{k^*}]^+, \quad (8)$$

and from (4) and (6) it follows that

$$\begin{aligned} k^* &= \min\{k; y_k = \max_{j \in \hat{K}} y_j\} = \min\{k; i_k + s_k = \max_{j \in \hat{K}} (i_j + s_j)\} \\ C(n) &= [y_{k^*}]^+ + \sum_{k \in \hat{K}} (s_k + r_k) = \max\{i_{k^*} + s_{k^*}, \sum_{k \in \hat{K}} (s_k + r_k)\}, \quad \text{for } n \geq 2. \quad (9) \end{aligned}$$

Equations (8) and (9) show that (3) and (2) hold.

Equation (8) implies that the sum of waiting times in the n -th cycle for $n \geq 2$ is constant and independent of n , and that

$$W_k(n) = 0 \quad \text{for all } k > k^* \text{ and } n \geq 2. \quad (10)$$

Suppose that $y_{k^*} > 0$. Then for $n \geq 2$ and $k \leq k^*$,

$$\begin{aligned} W_k(n) &= A_k(n) - A_{k-1}(n) \\ &= [(n-1)y_{m(k)} - (n-2)y_{k^*}]^+ - [(n-1)y_{m(k-1)} - (n-2)y_{k^*}]^+ \\ &= \begin{cases} (n-1)(y_{m(k)} - y_{m(k-1)}) & \text{if } y_{m(k-1)} > \frac{n-2}{n-1}y_{k^*}, \\ (n-1)y_{m(k)} - (n-2)y_{k^*} & \text{if } y_{m(k)} > \frac{n-2}{n-1}y_{k^*} \geq y_{m(k-1)}, \\ 0 & \text{if } \frac{n-2}{n-1}y_{k^*} \geq y_{m(k)}. \end{cases} \end{aligned}$$

Hence for $k < k^*$,

$$W_k(n) = 0 \text{ if } n \geq \frac{2y_{k^*} - y_{m(k)}}{y_{k^*} - y_{m(k)}} \text{ or } y_{m(k)} = y_{m(k-1)}.$$

Since $f(x) = \frac{2a-x}{a-x}$ is increasing in $x \in (-\infty, a)$ when $a > 0$, $\frac{2y_{k^*} - y_{m(k^*-1)}}{y_{k^*} - y_{m(k^*-1)}} \geq \frac{2y_{k^*} - y_{m(k)}}{y_{k^*} - y_{m(k)}}$ for $k = 1, \dots, k^* - 1$, and hence

$$W_k(n) = 0 \quad \text{for } k < k^*, \quad \text{if } n \geq \frac{2y_{k^*} - y_{m(k^*-1)}}{y_{k^*} - y_{m(k^*-1)}} \text{ or } y_{m(k)} = y_{m(k-1)}. \quad (11)$$

Equations (8) through (11) imply that if $y_{k^*} > 0$, then

$$W_{k^*}(n) = y_{k^*}, \quad W_k(n) = 0 \text{ for } k \neq k^*, \quad \text{for all } n \geq \frac{2y_{k^*} - y_{m(k^*-1)}}{y_{k^*} - y_{m(k^*-1)}}. \quad (12)$$

Therefore, we have the following proposition:

Proposition 1. When $y_{k^*} \leq 0$, the worker does not wait at any machine. The cycle time is $\sum_{k \in \hat{K}} (s_k + r_k)$, and if this does not meet the demand rate, then it is necessary to increase the number of workers. When $y_{k^*} > 0$, after the cycle given by (12) the worker waits y_{k^*} only at machine k^* , the cycle time is $i_{k^*} + s_{k^*}$, and the throughput, or the production rate of the U-shaped production line, is $\frac{1}{i_{k^*} + s_{k^*}}$. If this does not meet the demand rate, it is necessary to improve the production line by reducing the operation time s_{k^*} or replacing the bottleneck machine k^* with a new machine with smaller $i + s$. ■

We give a numerical example. The parameters are given as follows:

$$K = 4, \quad r_k = 1 \text{ for all } k, \quad (i_1, i_2, i_3, i_4) = (16, 19, 22, 17), \quad (s_1, s_2, s_3, s_4) = (2, 2, 1, 3).$$

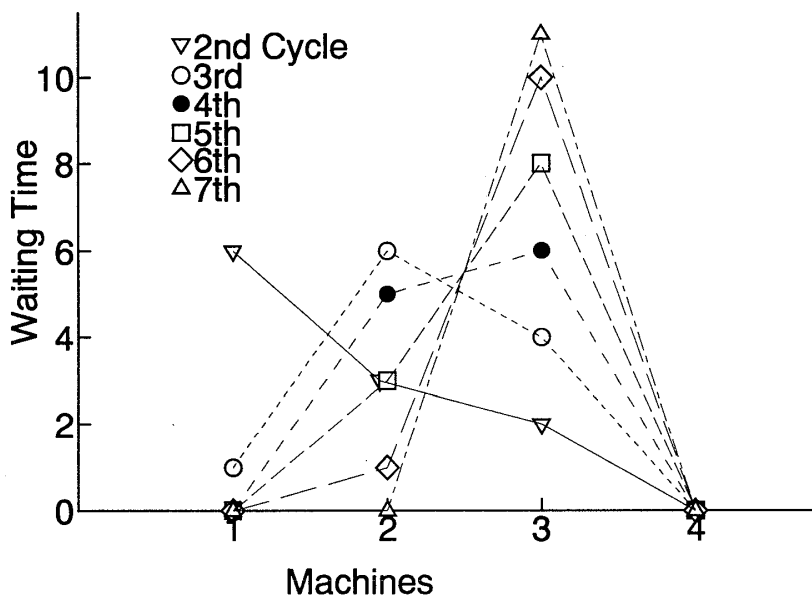


Figure 4. Example: Waiting Times of the Worker at Machines

Then $(y_1, y_2, y_3, y_4) = (6, 9, 11, 8)$, which implies $k^* = 3$. The waiting times of the worker at each machine for the 2nd, ..., 7th cycles are shown in Figure 4. As the number of cycle increases, the waiting time at the bottleneck machine 3 increases by 2, which is the value of $y_3 - (y_1 \vee y_2)$, and then becomes constant 11 after the 7th cycle, as shown by (12).

3. Optimal Worker Allocation Problem

In this section, we consider a production line with I workers and K machines, where $I \leq K$. For $i \in \hat{I} = \{1, 2, \dots, I\}$, we denote by \hat{K}_i the set of all machines which worker i operates, where for empty set ϕ , $\hat{K} = \cup_{i=1}^I \hat{K}_i$, $\hat{K}_i \cap \hat{K}_j = \phi$ for $i \neq j$, $i, j \in \hat{I}$, and \hat{K}_i may be empty. This allocation of I workers to K machines is denoted by $(\hat{K}_1, \hat{K}_2, \dots, \hat{K}_I)$. We derive the overall cycle time of this line, $C(\hat{K}_1, \hat{K}_2, \dots, \hat{K}_I)$. This reciprocal gives the throughput of the line.

Define $Y = \max_{k \in \hat{K}}(i_k + s_k)$, $S(\hat{K}_i) = \sum_{k \in \hat{K}_i} s_k$ for $i \in \hat{I}$ such that $\hat{K}_i \neq \phi$, and $S(\phi) = 0$. We

also denote the cycle time of worker $i \in \hat{I}$ and the sum of walking times of worker i in one cycle by $C(\hat{K}_i)$ and $R(\hat{K}_i)$, respectively. For example, if machines are arranged according to technical precedence relations, then $R(\hat{K}_i)$ is given by the total walking time of the route of worker i determined by the relations. Otherwise, $R(\hat{K}_i)$ may be given by the shortest of the total walking times in one cycle among all the possible routes of worker i . Then (9) leads to

$$C(\hat{K}_i) = \{\max_{k \in \hat{K}_i}(i_k + s_k)\} \vee \{S(\hat{K}_i) + R(\hat{K}_i)\}.$$

If there is a buffer for one item between any pair of machines where different workers operate, then the cycles of all workers can be synchronized. That is, for all $n \in \mathcal{Z}^+$, whenever the worker with the maximal cycle time ends his n -th cycle time, the other workers have already completed their n -th cycle times. Therefore, the overall cycle time of this line is equal to the maximal cycle time. That is,

$$C(\hat{K}_1, \hat{K}_2, \dots, \hat{K}_I) = \max_{i \in \hat{I}} C(\hat{K}_i) = Y \vee \max_{i \in \hat{I}} (S(\hat{K}_i) + R(\hat{K}_i)). \quad (13)$$

We consider an optimal worker allocation problem that minimizes the overall cycle time of the production line with I workers and K machines. Then this problem is formulated directly into the following optimization problem:

$$\begin{aligned} & \text{minimize} && T \\ & \text{subject to} && \sum_{i \in \hat{I}} x_{ij} = 1 \quad \text{for } j \in \hat{K}, \\ & && \sum_{j \in \hat{K}} s_j x_{ij} + R(\hat{K}_i(\{x_{ik}\})) \leq T \quad \text{for } i \in \hat{I}, \\ & && x_{ij} = 0 \text{ or } 1, \quad \text{for } i \in \hat{I}, j \in \hat{K}, \end{aligned} \quad (14)$$

where $\hat{K}_i(\{x_{ik}\}) = \{k \in \hat{K}; x_{ik} = 1\}$ for $i \in \hat{I}$. The variable x_{ij} takes the value 1 if worker i operates machine j , and 0, otherwise. The first constraint implies that each machine should be operated by just one worker, and the second one implies that the sum of operation and walking times of each worker in one cycle must be less than or equal to T . If the route of worker i is determined by the technical precedence relations, that is, worker i can move from machine k to machine k' with $(k, k') \in A_i$, then conditions (14) become

$$\sum_{i \in \hat{I}} x_{ik} = 1 \quad \text{for } k \in \hat{K},$$

$$\begin{aligned}
 &\sum_{k \in \hat{K}} s_k x_{ik} + \sum_{(k,k') \in A_i} r_{k,k'} y_{ikk'} \leq T \quad \text{for } i \in \hat{I}, \\
 &\sum_{(k,k') \in A_i} y_{ikk'} = x_{ik} \quad \text{for } k \in \hat{K} \quad \text{for } i \in \hat{I}, \\
 &\sum_{(k',k) \in A_i} y_{ik'k} = x_{ik} \quad \text{for } k \in \hat{K} \quad \text{for } i \in \hat{I}, \\
 &\sum_{(k,k') \in A_i} y_{ikk'} \leq |W| - 1 \quad \text{for all } W \subset \{k; x_{ik} = 1\}, W \neq \phi, W \neq \{k; x_{ik} = 1\}, \quad (15) \\
 &x_{ik} = 0 \text{ or } 1 \quad \text{for } k \in \hat{K}, i \in \hat{I}, \\
 &y_{ikk'} = 0 \text{ or } 1 \quad \text{for } (k, k') \in A_i, i \in \hat{I},
 \end{aligned}$$

where $r_{kk'}$ is the walking time from machine k to machine k' for $k, k' \in \hat{K}$ and $|S|$ denotes the number of components of set S . The variable $y_{ikk'}$ takes the value 1 if worker i walks from machine k to machine k' , and 0, otherwise. Inequality (15) creates just one route consisting of all elements of the set $\{k; x_{ik} = 1\}$ in the same way as the formulation of traveling salesman problems. Let T_I^* and C_I^* be a minimum objective value of this problem and a minimum overall cycle time of the line with I workers, respectively. It is clear that $C_I^* = Y \vee T_I^*$.

Although it is important to develop an algorithm for solving (14), it will not be discussed in this paper. Instead of it, we discuss the optimal worker allocation problem with one or two workers in the remainder of this section. This will reveal advantages of the U-shaped layout over the linear layout. Suppose that the daily demand, D , and the daily operating time, OT , are given. The problem is to find an optimal worker allocation with the minimum number of workers that can produce D in OT . Then (13) and (14) imply the following results:

- i) If $Y \geq S(\hat{K}) + R(\hat{K})$, then $C_1^* = Y$. If $\frac{OT}{C_1^*} \geq D$, then the allocation of one worker to all machines is optimal. Otherwise, the bottleneck machine attaining Y must be improved.
- ii) If $Y < S(\hat{K}) + R(\hat{K})$, then $C_1^* = S(\hat{K}) + R(\hat{K})$. If $\frac{OT}{C_1^*} \geq D$, then the allocation of one worker to all machines is optimal. Otherwise, we have to solve (14) with $I = 2$.

To obtain more concrete results from (14) with $I = 2$, we assume that $K = 4l$ for some $l \in \mathcal{Z}^+$, $s_k = s$ for all $k \in \hat{K}$, the machines are laid out in a U-shaped layout shown in Figure 5 and that the walking time between two machines is proportional to the distance between them. Note that since each operation mainly consists of detaching a processed item

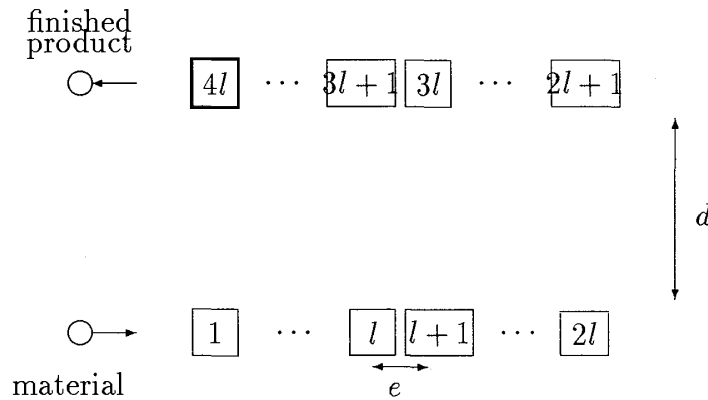
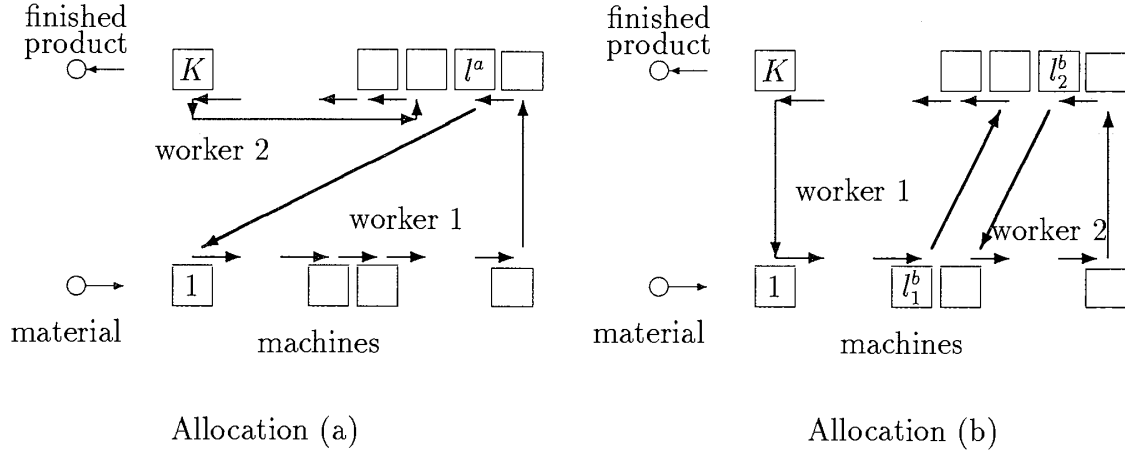


Figure 5. A U-shaped Layout with $4l$ Machines


Figure 6. Allocations of Two Workers

from the machine and attaching a new item to the machine, the above assumption $s_k = s$ for all $k \in \hat{K}$ will hold in many situations, and that machines are placed at the same distance when there are K machines of similar sizes. We denote the walking time between machines k and $k+1$ for $k \neq 2l, 4l$ by e , and that between machines k and $4l-k+1$ for $k \in \hat{K}$ by d . Then the walking time r_{jk} is given by

$$r_{jk} = \begin{cases} |j-k|e & \text{if } 1 \leq j, k \leq 2l \text{ or } 2l+1 \leq j, k \leq 4l, \\ \sqrt{d^2 + (4l-k-j+1)^2}e & \text{if } 1 \leq j \leq 2l < k \leq 4l, \text{ or } 1 \leq k \leq 2l < j \leq 4l. \end{cases}$$

Since the operation times of all machines are the same, Figure 5 implies that candidates for an optimal allocation can be restricted to the following two types of allocations (see Figure 6):

Allocation (a): worker 1 handles machines in $\hat{K}_1^a(l^a)$ and worker 2 handles machines in $\hat{K}_2^a(l^a)$, where $\hat{K}_1^a(l^a) = \{1, \dots, l^a\}$ and $\hat{K}_2^a(l^a) = \{l^a+1, \dots, K\}$ for some $1 \leq l^a < K$.

Allocation (b): worker 1 handles machines in $\hat{K}_1^b(l_1^b, l_2^b)$ and worker 2 handles machines in $\hat{K}_2^b(l_1^b, l_2^b)$, where $\hat{K}_1^b(l_1^b, l_2^b) = \{1, \dots, l_1^b, l_2^b+1, \dots, K\}$ and $\hat{K}_2^b(l_1^b, l_2^b) = \{l_1^b+1, \dots, l_2^b\}$ for some $1 \leq l_1^b < l_2^b < K$.

For allocation (a), if $l^a \geq 2l$, then $S(\hat{K}_1^a(l^a)) \geq 2ls$ and $R(\hat{K}_1^a(l^a)) \geq 2(2l-1)e$. Therefore from the symmetric property of the layout, $l^{a*} = 2l$ minimizes $C(\hat{K}_1^a(l^a), \hat{K}_2^a(l^a))$. Similarly, for allocation (b), $(l_1^{b*}, l_2^{b*}) = (l, 3l)$ minimizes $C(\hat{K}_1^b(l_1^b, l_2^b), \hat{K}_2^b(l_1^b, l_2^b))$. Hence the minimum cycle times of allocations of (a) and (b) are given by

$$\begin{aligned} C^{(a)} &= Y \vee \{2ls + 2(2l-1)e\} \\ \text{and } C^{(b)} &= Y \vee \{2ls + 2(l-1)e + 2d\}, \end{aligned}$$

respectively. If $Y \geq 2ls + 2(l-1)e + 2(le \vee d)$, then $C^{(a)} = C^{(b)} = Y$, and otherwise, $C^{(a)} \geq C^{(b)}$ if and only if $le \geq d$. Hence we obtain the following results:

i) If $Y \geq 2ls + 2(l-1)e + 2(le \vee d)$, then the minimum cycle time is Y , which is attained by $\hat{K}^{a*} = (\{1, \dots, 2l\}, \{2l+1, \dots, 4l\})$ or $\hat{K}^{b*} = (\{1, \dots, l, 3l+1, \dots, 4l\}, \{l+1, \dots, 3l\})$.

ii) If $2ls + 2(l-1)e + 2le > Y$ and $le \geq d$, then the minimum cycle time is $C^{(b)}$, which is attained by \hat{K}^{b*} .

iii) If $2ls + 2(l-1)e + 2d > Y$ and $le < d$, then the minimum cycle time is $C^{(a)}$, which is attained by \hat{K}^{a*} .

It usually holds in the U-shaped layout that le is larger than d . Therefore, allocation \hat{K}^{b*} attains the smaller cycle time than \hat{K}^{a*} .

We compare the overall cycle time of allocation $(\hat{K}_1^a(l^a), \hat{K}_2^a(l^a))$ in the U-shaped layout with that in a linear layout where $r_{jk} = |j - k|e$ for all $j, k \in \hat{K}, j \neq k$. From the symmetric property of the U-shaped and linear layouts, it can be assumed without loss of generality that $l^a \geq K/2$. If $d = e$, then the walking time from machine l^a to machine 1 in the U-shaped layout is less than or equal to that of the linear layout, and the other walking times are the same. Therefore, from (13) the overall cycle time of the U-shaped layout is less than or equal to that of the linear layout. Thus, we have the following proposition:

Proposition 2. When two workers are available in a production line with $K = 4l$ stations placed at the same distance and the same operation times, the U-shaped layout with allocation \hat{K}^{b*} has the smallest overall cycle time among the U-shaped layout with two types of worker allocations \hat{K}^a and \hat{K}^b and the linear layout.

4. Stochastic Case

In Section 2, for the U-shaped production line with a single worker we have analyzed his waiting time and a cycle time of this line, for the case where the operation, walking and processing times are constants. If multiple kinds of items are processed in this line, the processing and operation times are not constants. In addition, the operation and walking times of the worker may fluctuate because of his weariness and the learning effect. In this section, we deal with the U-shaped line with a single worker when the operation, walking and processing times are stochastic.

We define the following random variables: for $k \in \hat{K}$ and $n \in \mathcal{Z}^+$,
 $I_k(n)$: the processing time at machine k in the n -th cycle,
 $S_k(n)$: the operation time of the worker at machine k in the n -th cycle,
 $R_k(n)$: the walking time from machine k to machine $k + 1$ (K to 1, if $k = K$) in the n -th cycle.

In the same way as in Section 2, the initial condition implies that

$$W_k(1) = 0 \text{ for } k \in \hat{K}, \quad \text{and } C(1) = \sum_{k \in \hat{K}} (S_k(1) + R_k(1)).$$

For $k \in \hat{K}$ and $n \geq 2$, we have

$$A_k(n) = \max_{1 \leq j \leq k} [Y_j(n) + A_j(n-1) - A_K(n-1)]^+ = \max_{1 \leq j \leq k} [Y_j(n) - \sum_{l=j+1}^K W_l(n-1)]^+, \quad (16)$$

where

$$Y_k(n) = I_k(n-1) + S_k(n-1) - \sum_{l < k} (S_l(n) + R_l(n)) - \sum_{l \geq k} (S_l(n-1) + R_l(n-1)). \quad (17)$$

The proof of (16) is given in Appendix A. For $n \geq 2$, we define $k^*(n)$ as

$$k^*(n) = \min \left\{ k \in \hat{K}; [Y_k(n) - \sum_{l=k+1}^K W_l(n-1)]^+ = \max_{j \in \hat{K}} [Y_j(n) - \sum_{l=j+1}^K W_l(n-1)]^+ \right\}.$$

Then (16) implies that

$$W_1(2) = [Y_1(2)]^+, \quad A_k(2) = \max_{1 \leq j \leq k} [Y_j(2)]^+ \quad \text{for } k \in \hat{K},$$

$$A_k(2) = A_{k^*(2)}(2) \text{ for } k \geq k^*(2) \text{ and } W_k(2) = 0 \text{ for } k > k^*(2).$$

Similarly, for $n \geq 3$

$$A_k(n) = A_{k^*(n)}(n) \text{ for } k \geq k^*(n) \text{ and } W_k(n) = 0 \text{ for } k > k^*(n). \quad (18)$$

Equations (16), (17) and (18) correspond to (7), (4) and (10) in Section 2, respectively.

In the following, we assume the following condition:

Condition (C). For some $k^* \in \hat{K}$,

$$P([Y_{k^*}(2)]^+ > [Y_j(2)]^+ \text{ for all } j < k^*, [Y_{k^*}(2)]^+ \geq [Y_j(2)]^+ \text{ for all } j > k^*) = 1, \quad (19)$$

and $\{I_k(n); n \in \mathcal{Z}^+\}$, $\{S_k(n); n \in \mathcal{Z}^+\}$ and $\{R_k(n); n \in \mathcal{Z}^+\}$ are sequences of independent and identically distributed random variables with finite means i_k , s_k and r_k for $k \in \hat{K}$, respectively, and are mutually independent. ■

Under Condition (C), since $\{(Y_1(n), \dots, Y_K(n)); n = 2, 3, \dots\}$ is stationary, for all $n \geq 2$,

$$[Y_{k^*}(n)]^+ > [Y_j(n)]^+ \text{ for all } j < k^*, \text{ and } [Y_{k^*}(n)]^+ \geq [Y_j(n)]^+ \text{ for all } j > k^* \quad (20)$$

hold with probability one (w. p. 1). Since $Y_j(n)$ denotes the difference between the $(n - 1)$ -th processing time and the time interval from the $(n - 1)$ -th departure to the n -th arrival at machine j under the assumption of no waiting times, equation (19) means that if $Y_{k^*}(n)$ is positive, then it is larger than $Y_j(n)$ at all machines $j < k^*$ and larger than or equal to $Y_j(n)$ at all machines $j \geq k^*$. We consider the properties of the waiting time process $\{W_k(n); k \in \hat{K}, n \in \mathcal{Z}^+\}$ and the cycle time process $\{C(n); n \in \mathcal{Z}^+\}$ under Condition (C). Note that Condition (C) is satisfied when the processing, operation and walking times are constants as in Sections 2 and 3.

From (20), it follows that

$$k^*(n) = k^* \text{ for all } n \geq 2, \text{ w.p.1.} \quad (21)$$

The proof of (21) is given in Appendix B. Equations (16), (18) and (21) imply that for $n \geq 2$,

$$A_k(n) = [Y_{k^*}(n)]^+ \text{ for all } k \geq k^*, \text{ and } W_k(n) = 0 \text{ for all } k > k^*. \quad (22)$$

That is, the worker does not wait for the completion of processing at machines $k^* + 1, \dots, K$. The n -th cycle time for $n \geq 2$ is given by

$$\begin{aligned} C(n) &= A_K(n) + \sum_{k \in \hat{K}} (S_k(n) + R_k(n)) = [Y_{k^*}(n)]^+ + \sum_{k \in \hat{K}} (S_k(n) + R_k(n)) \\ &= \left\{ \sum_{k \in \hat{K}} (S_k(n) + R_k(n)) \right\} \vee \{I_{k^*}(n - 1) + S_{k^*}(n - 1) \\ &\quad + \sum_{l \geq k^*} (S_l(n) + R_l(n) - S_l(n - 1) - R_l(n - 1))\}. \end{aligned} \quad (23)$$

Consequently, for $n \geq 2$ the n -th cycle time has an identical probability distribution which is independent of n , and the expected n -th cycle time is given by

$$E[C(n)] = E[[Y_{k^*}(n)]^+] + \sum_{k \in \hat{K}} (s_k + r_k). \quad (24)$$

Equations (22) and (23) correspond to (8) and (9) in Section 2, respectively.

When $k^* = 1$ and $P(Y_1(2) \leq 0) > 0$, (22) implies that the worker waits for the end of processing at machine 1 in the n -th cycle if and only if $Y_1(n) > 0$, and that he does not wait at the other machines in any cycle. Therefore, if $k^* = 1$ and $P(Y_1(2) > 0) = 1$ hold, then the worker waits for the completion of processing at machine 1 in all cycles. In the following, we assume that $k^* \geq 2$ holds. Since the first inequality of (20) implies that $Y_{k^*}(n) > 0$ w.p. 1 for $n \geq 2$, equations (22) through (24) are reduced to

$$\begin{aligned} A_{k^*}(n) &= Y_{k^*}(n), \quad W_k(n) = 0 \quad \text{w. p. 1 for all } k > k^*, \\ C(n) &= I_{k^*}(n-1) + S_{k^*}(n-1) + \sum_{l \geq k^*} (S_l(n) + R_l(n) - S_l(n-1) - R_l(n-1)) \quad \text{w. p. 1,} \end{aligned}$$

and

$$E[C(n)] = i_{k^*} + s_{k^*}, \quad \text{for } n \geq 2. \quad (25)$$

Equation (16) implies that

$$W_{k^*}(n) = A_{k^*}(n) - A_{k^*-1}(n)$$

$$\begin{aligned} &= Y_{k^*}(n) - \max_{1 \leq j \leq k^*-1} [Y_j(n) - \sum_{l=j+1}^{k^*} W_l(n-1)]^+ \\ &= Y_{k^*}(n) - \left\{ \left[\max_{1 \leq j \leq k^*-1} \left\{ Y_j(n) - \sum_{l=j+1}^{k^*-1} W_l(n-1) \right\} \right] \vee W_{k^*}(n-1) - W_{k^*}(n-1) \right\}, \quad \text{w. p. 1.} \end{aligned}$$

Therefore,

$$\begin{aligned} &W_{k^*}(n) - W_{k^*}(n-1) \\ &= \min_{1 \leq j \leq k^*-1} \left\{ Y_{k^*}(n) - Y_j(n) + \sum_{l=j+1}^{k^*-1} W_l(n-1) \right\} \wedge \{Y_{k^*}(n) - W_{k^*}(n-1)\}, \quad \text{w. p. 1.} \end{aligned}$$

If $Y_{k^*}(n) > W_{k^*}(n-1)$, then (20) implies

$$W_{k^*}(n) - W_{k^*}(n-1) \geq \min_{1 \leq j \leq k^*-1} \{Y_{k^*}(n) - [Y_j(n)]^+\} \wedge \{Y_{k^*}(n) - W_{k^*}(n-1)\} > 0.$$

If $Y_{k^*}(n) \leq W_{k^*}(n-1)$, then from (16), (20) and (22),

$$A_{k^*-1}(n) = \max_{1 \leq j \leq k^*-1} [Y_j(n) - \sum_{l=j+1}^{k^*} W_l(n-1)]^+ \leq [Y_{k^*}(n) - W_{k^*}(n-1)]^+ = 0,$$

which implies by (22) that $W_j(n) = 0$ for $j < k^*$ and $A_{k^*}(n) = W_{k^*}(n) = Y_{k^*}(n)$. That is, the waiting time process $\{W_k(n); k \in \hat{K}, n \in \mathcal{Z}^+\}$ behaves as follows: for $n \geq 2$,

- i) if $Y_{k^*}(n) > W_{k^*}(n-1)$, then $W_{k^*}(n) > W_{k^*}(n-1)$ and $W_j(n) = 0$ for all $j > k^*$,
- ii) if $Y_{k^*}(n) = W_{k^*}(n-1)$, then $W_{k^*}(n) = Y_{k^*}(n) = W_{k^*}(n-1)$ and $W_k(n) = 0$ for all $k \neq k^*$,
- iii) if $Y_{k^*}(n) < W_{k^*}(n-1)$, then $W_{k^*}(n) = Y_{k^*}(n) < W_{k^*}(n-1)$ and $W_k(n) = 0$ for all $k \neq k^*$.

From the above behavior i) to iii) and (25), we have the following proposition:

Proposition 3. Under Condition (C) and $k^* \geq 2$, the worker always waits at the bottleneck machine k^* , and he may wait at machine $j < k^*$ in the n -th cycle only if $W_{k^*}(n) > W_{k^*}(n-1)$.

The throughput of this line is given by $\frac{1}{i_{k^*} + s_{k^*}}$. ■

If the throughput $\frac{1}{i_{k^*} + s_{k^*}}$ does not meet the demand rate, it is necessary to improve the bottleneck machine.

Remark. For the U-shaped line with multiple workers, the cycle times of each worker fluctuate with cycles. That is, in one cycle he may force the others to wait for the end of his cycle, and in another cycle some other worker may force him to do. Therefore, the expected overall cycle time of this line is not always the maximum of the expected cycle times of workers.

5. Concluding Remarks

In this paper, we first considered the U-shaped production line with one multi-function worker, and obtained his waiting time and the cycle time for the case where the processing, operation and walking times are constants. Then we derived the overall cycle time of the U-shaped line with I multi-function workers, and formulated the optimal worker allocation problem that minimizes the overall cycle time. We also investigated the problem with $I = 1$ or 2 in the production line with K machines placed at the same distance, and showed the advantages of the U-shaped layout over the linear layout. Finally, we discussed the U-shaped production line with one worker in which the processes of operation, walking and processing times are stochastic. We showed that the worker waits for the completion of processing at a bottleneck machine in all cycles under Condition (C).

In the future research we wish to investigate stochastic properties of the cycle time and the waiting time for the U-shaped production line in which Condition (C) does not hold. An efficient algorithm to optimize worker allocation would also be desirable.

Acknowledgments. The authors would like to express their appreciation to an associate editor and anonymous referees for their helpful comments.

Appendix A. Proof of Equation (16)

Since the time from the $(n - 1)$ -th departure to the n -th arrival of the worker at machine k is given by

$$R_k(n - 1) + \sum_{j>k} (W_j(n - 1) + S_j(n - 1) + R_j(n - 1)) + \sum_{j<k} (W_j(n) + S_j(n) + R_j(n)),$$

the waiting time at machine k is given by

$$\begin{aligned} W_k(n) &= \left[I_k(n - 1) - \left\{ R_k(n - 1) + \sum_{j>k} (W_j(n - 1) + S_j(n - 1) + R_j(n - 1)) \right. \right. \\ &\quad \left. \left. + \sum_{j<k} (W_j(n) + S_j(n) + R_j(n)) \right\} \right]^+ \\ &= \left[Y_k(n) - \sum_{j<k} W_j(n) - \sum_{j>k} W_j(n - 1) \right]^+, \quad \text{for } k \in \hat{K}, n \geq 2, \end{aligned} \tag{A1}$$

where $Y_k(n)$ is given by (17). First, it follows from (A1) and (17) that for $n \geq 2$,

$$A_1(n) = W_1(n) = [Y_1(n) - \sum_{j>1} W_j(n - 1)]^+ = [Y_1(n) + A_1(n - 1) - A_K(n - 1)]^+.$$

In addition,

$$\begin{aligned}
A_2(n) &= A_1(n) + W_2(n) \\
&= A_1(n) + [Y_2(n) - A_1(n) - (A_K(n-1) - A_2(n-1))]^+ \\
&= \{Y_2(n) + A_2(n-1) - A_K(n-1)\} \vee A_1(n) \\
&= \max_{1 \leq j \leq 2} [Y_j(n) + A_j(n-1) - A_K(n-1)]^+.
\end{aligned}$$

Repeating this process, we derive (16).

Appendix B: Proof of Equation (21)

Condition (C) implies that $k^*(2) = k^*$, w.p.1. Suppose that $k^*(n) = k^*$, w. p. 1 for $n \geq 2$. Then it follows from (18) and (20) that

$$P(k^*(n+1) = k^*) = P(k^*(n+1) = k^* | k^*(n) = k^*)P(k^*(n) = k^*)$$

$$\begin{aligned}
&= P([Y_{k^*}(n+1)]^+ > \max_{1 \leq j \leq k^*-1} [Y_j(n+1) - \sum_{l=j+1}^{k^*} W_l(n)]^+, \\
&\quad \text{and } [Y_{k^*}(n+1)]^+ \geq \max_{k^*+1 \leq j \leq K} [Y_j(n+1)]^+ | k^*(n) = k^*)P(k^*(n) = k^*) \\
&\geq P([Y_{k^*}(n+1)]^+ > \max_{1 \leq j \leq k^*-1} [Y_j(n+1)]^+, \text{ and} \\
&\quad [Y_{k^*}(n+1)]^+ \geq \max_{k^*+1 \leq j \leq K} [Y_j(n+1)]^+ | k^*(n) = k^*)P(k^*(n) = k^*) \\
&= P([Y_{k^*}(n+1)]^+ > \max_{1 \leq j \leq k^*-1} [Y_j(n+1)]^+ \text{ and } [Y_{k^*}(n+1)]^+ \geq \max_{k^*+1 \leq j \leq K} [Y_j(n+1)]^+) \\
&= 1.
\end{aligned}$$

The proof of (21) is concluded.

References

- [1] Baybars, I., A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Management Science*, vol. 32, pp. 909-932, 1986.
- [2] Miltenburg, G. J. and Wijngaard, J., The U-line Line Balancing Problem, *Management Science*, vol. 40, pp. 1378-1388, 1994.
- [3] Monden, Y., *Toyota Production System: An Integrated Approach to Just-In-Time*, 2nd Edition, Industrial Engineering and Management Press, Georgia, 1993.

Koichi NAKADE
Department of Systems Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466 Japan
E-mail: nakade@system.nitech.ac.jp