Electronic Preprint for Journal of Information Processing Vol.24 No.1

Regular Paper

On the Redundancy of Delivery Time in an In-Line Machine Model

Eishi Chiba[1,a)]

Received: April 15, 2015, Accepted: August 12, 2015

Abstract: In this paper, we focus on an in-line machine model. This model represents systems for the manufacturing of a product in large quantities. Recently, studies relating to the collision probability between jobs have been conducted in such models. In this paper, we extend the known models to a generalized version by considering delivery time between machines. We first present a method for computing a schedule of jobs in the generalized model. Then, we show that the collision probability for the generalized model is the same as that for the model without delivery time. We call this property the redundancy of delivery time. Next, we introduce two optimization problems with collision probability for the generalized model. Using the redundancy of delivery time, it is shown that these optimization problems are equivalent to simpler problems. This finding may prove to be very useful when considering optimization problems with collision probability.

Keywords: production scheduling, operations research, in-line machine model, collision probability, delivery time

1. Introduction

In this paper, we focus on a simple model with applications in the manufacturing of a product in large quantities. The following model was introduced in Ref. [1]. The machines in this model exist in series. Each job is fed into the first machine from the entrance with a constant time interval between jobs. Each job is processed in order from the first machine to the last, and is finally delivered to the exit. Each machine can process only one job at a time. The processing time at each machine is stochastic. Buffers exist in front of each machine. These buffers are used as a temporary housing for jobs waiting to be processed. For simplicity, the delivery time between two machines is assumed to be nil.

In the above model, the phenomenon of a collision occurring between jobs is the main subject of observation. A collision is said to have occurred if a job is delivered to a machine when all buffers in front of the machine are occupied. The phenomenon of a collision occurring could cause the breakdown of a manufacturing system and, therefore, can be considered to be one of the most potentially damaging things that can happen in a production line. In particular, the probability of a collision occurring is an important evaluation item, and this has applications to the efficiency of manufacturing systems (see Refs. [1], [2] and [3]).

In the field of scheduling research, *blocking* exists in flow shops, this being similar to a collision. When considering blocking, the number of buffers is assumed to be limited (since if the number of buffers is unlimited, no blocking occurs). Even when a machine completes a job, if all buffers in front of the next machine are occupied, the job stays in the current machine. Therefore, the machine cannot process subsequent jobs. At such a time, the machine is said to be blocked, or the job (staying in the machine) is said to be blocked. Blocking is the phenomenon of a blocked machine or a blocked job occurring (see Ref. [12]). Note that the probability of a blocking occurring is equal to that of a collision occurring for the model in Ref. [1]. Even if a machine is blocked, the blocked job is delivered to the next machine as soon as a buffer in front of that next machine becomes empty. Under such blocking conditions, manufacturing systems are still able to continue operating. On the other hand, under collision conditions, manufacturing systems stop when a collision occurs. This concept is the difference between blocking and collision. Regarding deterministic processing time, the flow shop problem with blocking was studied in detail by Refs. [9], [11] and [13], where the purpose was to minimize the makespan. A survey paper on this problem is in Ref. [4]. A fair amount of research has also been done on the development of heuristics for this problem (one popular heuristics is in Ref. [10]). With regards to stochastic processing time, study results are somewhat limited in comparison to their deterministic counterparts. For example, see Refs. [6] and [11], where the purpose was to minimize the expected makespan.

In queueing theory, there is a tandem queue model, this being similar to the model in Ref. [1]. In the tandem queue model, depending on the rule for processing blockings (blocked calls cleared, blocked calls delayed, etc.), analyses of performance measures in the steady state are the main object of research. Examples of performance measures are the expected queue length, the expected waiting time, etc. Collision probability relates to loss probability, this being an important evaluation item in queueing theory. Note that, loss probability is the probability of a blocking occurring in the steady state, whereas collision probability is an evaluation item in an unsteady state. We can find, in

[1] Department of Industrial and Systems Engineering, Hosei University, Koganei, Tokyo 184–8584, Japan
[a)] e-chiba@hosei.ac.jp

fact, that a wide range of literature in the field of queuing theory has been addressed, for example in Refs. [5], [7] and [8].

For the model in Ref. [1], when buffers are not considered (i.e., the number of buffers is zero), previous results relating to the analysis of the collision probability exist. For example, it was shown in Ref. [2] that the collision probability can be approximately expressed by a multiple integration, assuming that the processing time of each machine follows general distribution. In the special case that the processing time follows exponential distribution, a closed form formula of the approximate collision probability is derived without using any multiple integration (see Ref. [2]). Moreover, if the processing time follows Erlang distribution, a closed form formula of the approximate collision probability is derived (see Ref. [3]). On the other hand, for the model in Ref. [1] (including buffers), although no analytical studies on the collision probability seem to exist, a computer simulation method for computing the collision probability was presented therein. This simulation method is applicable even when the processing time follows general distribution.

In this paper, we extend the model in Ref. [1] to a generalized version by considering delivery time between two machines. Since delivery time exists in real production lines, the generalized model seems to be reasonable. It is permitted to deliver an arbitrary number of jobs at one time (Note that, if it is assumed that we can deliver only one job at a time, then we can regard a delivery as a machine.). This assumption might be reasonable for applications that include delivery by machinery such as conveyor belts. In addition, the delivery time is assumed to be deterministic and independent of the jobs (Note that the delivery time is not stochastic.).

First, we present a method for computing a schedule of jobs in the generalized model. Using this schedule, we can compute the collision probability by a computer simulation method. Concretely speaking, by applying the idea of the simulation method in Ref. [1], we can compute the probability. In addition, we show that the collision probability can be computed without data on the delivery time. This means that the delivery time is not an essential parameter when computing collision probability. We call this property *the redundancy of delivery time*. Due to the redundancy of delivery time, collision probability can be computed by a simpler method. Next, we introduce two optimization problems for the generalized model. The two optimization problems with collision probability were originally presented in Refs. [1] and [2]. We rewrite these two optimization problems as the two optimization problems for the generalized model. Moreover, it is shown that we can get more understanding of those problems from the redundancy of delivery time. The discussion in Section 5 of this paper might be very useful when considering optimization problems with collision probability.

2. In-line Machine Model

We describe an in-line machine model with delivery time. This is a generalized version of the model in Ref. [1], with consideration of delivery time. The following notations are used:

- $M_1, M_2, \ldots, M_m$: m machines, which line in series.
- $J_1, J_2, \ldots, J_n$: n jobs to be processed.

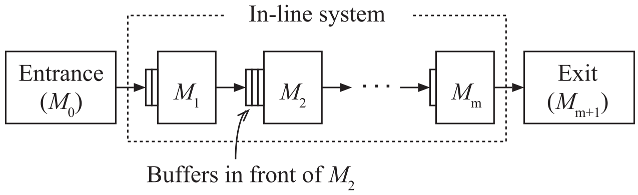

Fig. 1 Example of the in-line machine model.

- $T_i^{(j)}$ (> 0): Processing time of J_i at M_j.
- t_{tact} (> 0): Tact time, i.e., the time difference between the start time instants of J_i and J_{i+1} for all $1 \le i \le n-1$ at the entrance to the line.
- $b^{(j)}$ ($\in \mathbb{Z}_+$): The number of buffers in front of M_j, where $\mathbb{Z}_+$ denotes the nonnegative integer set.
- $d^{(j)}$ (≥ 0): Delivery time between M_j and M_{j+1}.

The in-line machine model is illustrated in **Fig. 1**. With the same time interval, t_{tact}, jobs are fed one by one into the line at the entrance. Each job J_i is fed i-th into the line. Each job is first processed on a machine M_1. It is then automatically delivered to the next machine M_2 after it has been finished on M_1. The delivery time between M_1 and M_2 is $d^{(1)}$. As soon as M_2 receives the job, it starts processing. In this manner, each job is processed on consecutive machines in the order $M_1, M_2, \ldots, M_m$, and then sent to the exit. For convenience of notation, the entrance and exit are denoted by M_0 and M_{m+1}, respectively. Moreover, the processing time $T_i^{(j)}$ is assumed to be a random variable. The delivery time $d^{(j)}$ is assumed to be deterministic and independent of the jobs.

When job J_i arrives at M_j, the system dynamics are as follows. If M_j is idle, it starts processing J_i. If M_j is processing another job and an empty buffer exists in front of M_j, then J_i waits at the buffer. When waiting, the queue discipline follows First-In-First-Out (FIFO), where all jobs are processed in the same order as they arrive in the queue. Since the number of buffers in front of M_j is b_j, the length of the queue can be, at most, b_j. In Fig. 1, $b_1 = 2$, $b_2 = 3$, and $b_m = 1$. When all buffers in front of M_j are occupied (i.e., the length of the queue in front of M_j is b_j), if J_i arrives at M_j, then a *collision* at M_j occurs. The *collision probability* is the probability that there will be at least one collision at any machine.

3. Scheduling When the Number of Buffers Is Infinite

We consider a schedule under the assumption that the number of buffers in front of each machine is infinite. In this paper, a *schedule* is defined by the three functions a, s, and f. Each function is defined as follows:

- $a_i^{(j)}$: Time instant when job J_i arrives at M_j,
- $s_i^{(j)}$: Time instant when job J_i is started on M_j,
- $f_i^{(j)}$: Time instant when job J_i is finished on M_j.

From the assumption that the number of buffers in front of each machine is infinite, no collisions will occur. Therefore, the schedule $(a_i^{(j)}, s_i^{(j)}, f_i^{(j)} : 1 \le i \le n, 1 \le j \le m)$ can exist (Note that, if we do not make this assumption, a system stoppage could occur due to a collision, and after any such stoppage a schedule cannot exist.). Note that, when considering the schedule $(a_i^{(j)}, s_i^{(j)}, f_i^{(j)} : 1 \le i \le n, 1 \le j \le m)$, the number of buffers in front of each machine is assumed to be infinite.

In the concrete computation of the schedule, the random variable $T_i^{(j)}$ is generated randomly from a distribution. The variable $t_i^{(j)}$ is assumed to be the generated value derived from $T_i^{(j)}$. After the generation of $t_i^{(j)}$, we can regard $t_i^{(j)}$ as a constant. Given the number of jobs n, the number of machines m, the processing time $t_i^{(j)}$, the tact time t_{tact}, and the delivery time $d^{(j)}$, we consider the computation of an optimal schedule.

After J_i is finished on M_{j-1}, it arrives at M_j in $d^{(j-1)}$ unit time. J_i is started on M_j after the completion of two events, these being (i) J_i arrives at M_j and (ii) J_{i-1} is finished on M_j. After J_i is started on M_j, it is finished in $t_i^{(j)}$ unit time. Each J_i is assumed to be finished on M_0 at the time instant $(i-1)t_{\text{tact}}$ since each job is fed into the line at the entrance with t_{tact} time interval. Moreover, J_0 is assumed to be finished on M_j at time instant 0. Therefore, the following recursive expressions hold:

$$a_i^{(j)} = f_i^{(j-1)} + d^{(j-1)} \ (1 \le i \le n, 1 \le j \le m+1), \tag{1}$$

$$s_i^{(j)} = \max\left\{a_i^{(j)}, f_{i-1}^{(j)}\right\} \ (1 \le i \le n, 1 \le j \le m), \tag{2}$$

$$f_i^{(j)} = \begin{cases} s_i^{(j)} + t_i^{(j)} & (1 \le i \le n, 1 \le j \le m), \\ (i-1)t_{\text{tact}} & (1 \le i \le n, j = 0), \\ 0 & (i = 0, 1 \le j \le m). \end{cases} \tag{3}$$

From Eqs. (1), (2), and (3), the time complexity for computing the schedule is $O(mn)$, as filling each entry requires $O(1)$ time.

Definition 1 A waiting sequence in front of M_j is a sequence $(J_p, J_{p+1}, \ldots, J_q)$ for some $2 \le p \le q \le n$ such that

$$a_k^{(j)} < f_{p-1}^{(j)}$$

for all $k = p, p+1, \ldots, q$.
Note that the waiting sequence is defined under the assumption that the number of buffers is infinite.

Next, using the schedule, we explain that it is possible to determine whether a collision occurs or not in the in-line machine model with finite buffers. Namely, if the length of the waiting sequence in front of M_j is larger than the number of buffers $b^{(j)}$, a collision at M_j occurs in the in-line machine model with finite buffers.

4. Theorem

The recursive expressions on the schedule that does not consider delivery time are derived from the consideration of $d^{(j-1)} = 0$ in Eqs. (1), (2), and (3). Therefore, when the schedule that does not consider delivery time is denoted by $\tilde{a}$, $\tilde{s}$, and $\tilde{f}$, the following recursive expressions hold:

$$\tilde{a}_i^{(j)} = \tilde{f}_i^{(j-1)} \ (1 \le i \le n, 1 \le j \le m+1),$$

$$\tilde{s}_i^{(j)} = \max\left\{\tilde{a}_i^{(j)}, \tilde{f}_{i-1}^{(j)}\right\} \ (1 \le i \le n, 1 \le j \le m),$$

$$\tilde{f}_i^{(j)} = \begin{cases} \tilde{s}_i^{(j)} + t_i^{(j)} & (1 \le i \le n, 1 \le j \le m), \\ (i-1)t_{\text{tact}} & (1 \le i \le n, j = 0), \\ 0 & (i = 0, 1 \le j \le m). \end{cases}$$

We prove that the time difference between the two schedules (a, s, f) and $(\tilde{a}, \tilde{s}, \tilde{f})$ remains constant. In fact, we can prove that the three functions (a, s, and f) are shifted by a constant related to delivery time.

Lemma 1 For all i, j $(1 \le i \le n, 1 \le j \le m)$, the following proposition $P(i, j)$ holds:

$$a_i^{(j)} - \tilde{a}_i^{(j)} = s_i^{(j)} - \tilde{s}_i^{(j)} = f_i^{(j)} - \tilde{f}_i^{(j)} = \sum_{l=0}^{j-1} d^{(l)}.$$

Proof We prove it by double induction on i and j.
STAGE 1: $\forall i, P(i, 1)$ is proved. First, for all i, the following holds:

$$\begin{aligned}
&a_i^{(1)} - \tilde{a}_i^{(1)} \\
&= (f_i^{(0)} + d^{(0)}) - \tilde{f}_i^{(0)} \qquad \text{(from Eq. (1))} \\
&= ((i-1)t_{\text{tact}} + d^{(0)}) - (i-1)t_{\text{tact}} \qquad \text{(from Eq. (3))} \\
&= d^{(0)}. \\
&f_i^{(1)} - \tilde{f}_i^{(1)} \\
&= (s_i^{(1)} + t_i^{(1)}) - (\tilde{s}_i^{(1)} + t_i^{(1)}) \qquad \text{(from Eq. (3))} \\
&= s_i^{(1)} - \tilde{s}_i^{(1)}.
\end{aligned} \tag{4}$$

Next, we prove that $s_i^{(1)} - \tilde{s}_i^{(1)} = d^{(0)}$ for all i holds by induction on i. When $i = 1$, the following holds:

$$\begin{aligned}
&s_1^{(1)} - \tilde{s}_1^{(1)} \\
&= \max\{a_1^{(1)}, f_0^{(1)}\} - \max\{\tilde{a}_1^{(1)}, \tilde{f}_0^{(1)}\} \qquad \text{(from Eq. (2))} \\
&= a_1^{(1)} - \tilde{a}_1^{(1)} \qquad \text{(from Eq. (3))} \\
&= d^{(0)}. \qquad \text{(from Eq. (4))}
\end{aligned}$$

When $i \ge 2$, the following holds from Eq. (2).

$$s_i^{(1)} - \tilde{s}_i^{(1)} = \max\{a_i^{(1)}, f_{i-1}^{(1)}\} - \max\{\tilde{a}_i^{(1)}, \tilde{f}_{i-1}^{(1)}\}. \tag{5}$$

We consider four cases in Eq. (5).

(i) When $a_i^{(1)} \ge f_{i-1}^{(1)}, \tilde{a}_i^{(1)} \ge \tilde{f}_{i-1}^{(1)}$:

$$\begin{aligned}
\text{Eq. (5)} &= a_i^{(1)} - \tilde{a}_i^{(1)} \\
&= d^{(0)}. \qquad \text{(from Eq. (4))}
\end{aligned}$$

(ii) When $a_i^{(1)} \ge f_{i-1}^{(1)}, \tilde{a}_i^{(1)} < \tilde{f}_{i-1}^{(1)}$:

$$\begin{aligned}
&a_i^{(1)} \ge f_{i-1}^{(1)} \\
&\Longleftrightarrow \tilde{a}_i^{(1)} + d^{(0)} \ge s_{i-1}^{(1)} + t_{i-1}^{(1)} \quad \text{(from Eqs. (3) and (4))} \\
&\Longleftrightarrow \tilde{a}_i^{(1)} + d^{(0)} \ge (\tilde{s}_{i-1}^{(1)} + d^{(0)}) + t_{i-1}^{(1)} \\
&\qquad\qquad \text{(from the induction hypothesis)} \\
&\Longleftrightarrow \tilde{a}_i^{(1)} \ge \tilde{f}_{i-1}^{(1)} \qquad \text{(from Eq. (3))}
\end{aligned}$$

The last inequality contradicts the hypothesis in case (ii). Therefore, case (ii) is impossible.

(iii) When $a_i^{(1)} < f_{i-1}^{(1)}, \tilde{a}_i^{(1)} \ge \tilde{f}_{i-1}^{(1)}$: We can prove that case (iii) is impossible using the same method as used in case (ii).

(iv) When $a_i^{(1)} < f_{i-1}^{(1)}, \tilde{a}_i^{(1)} < \tilde{f}_{i-1}^{(1)}$:

$$\begin{aligned}
\text{Eq. (5)} &= f_{i-1}^{(1)} - \tilde{f}_{i-1}^{(1)} \\
&= (s_{i-1}^{(1)} + t_{i-1}^{(1)}) - (\tilde{s}_{i-1}^{(1)} + t_{i-1}^{(1)}) \qquad \text{(from Eq. (3))} \\
&= d^{(0)}. \qquad \text{(from the induction hypothesis)}
\end{aligned}$$

Then, the induction finishes.
STAGE 2: $\forall j, P(1, j)$ is proved. First, for all j, the following holds:

$$\begin{aligned}
&s_1^{(j)} - \tilde{s}_1^{(j)} \\
&= \max\{a_1^{(j)}, f_0^{(j)}\} - \max\{\tilde{a}_1^{(j)}, \tilde{f}_0^{(j)}\} \qquad \text{(from Eq. (2))}
\end{aligned}$$

$$= a_1^{(j)} - \tilde{a}_1^{(j)}. \qquad \text{(from Eq. (3))} \tag{6}$$

$$f_1^{(j)} - \tilde{f}_1^{(j)}$$
$$= (s_1^{(j)} + t_1^{(j)}) - (\tilde{s}_1^{(j)} + t_1^{(j)}) \qquad \text{(from Eq. (3))}$$
$$= s_1^{(j)} - \tilde{s}_1^{(j)}. \tag{7}$$

Next, we prove that $a_1^{(j)} - \tilde{a}_1^{(j)} = \sum_{l=0}^{j-1} d^{(l)}$ for all j holds by induction on j. When $j = 1$, the following holds:

$$a_1^{(1)} - \tilde{a}_1^{(1)}$$
$$= (f_1^{(0)} + d^{(0)}) - \tilde{f}_1^{(0)} \qquad \text{(from Eq. (1))}$$
$$= d^{(0)}. \qquad \text{(from Eq. (3))}$$

When $j \geq 2$, the following holds:

$$a_1^{(j)} - \tilde{a}_1^{(j)}$$
$$= (f_1^{(j-1)} + d^{(j-1)}) - \tilde{f}_1^{(j-1)} \qquad \text{(from Eq. (1))}$$
$$= (a_1^{(j-1)} - \tilde{a}_1^{(j-1)}) + d^{(j-1)} \qquad \text{(from Eqs. (6) and (7))}$$
$$= \sum_{l=0}^{j-2} d^{(l)} + d^{(j-1)} \qquad \text{(from the induction hypothesis)}$$
$$= \sum_{l=0}^{j-1} d^{(l)}.$$

STAGE 3: We prove that $P(i, j)$ holds if $P(i, j-1)$ and $P(i-1, j)$ hold for all i, j $(2 \leq i \leq n,\ 2 \leq j \leq m)$.

First, the following holds for all i, j.

$$a_i^{(j)} - \tilde{a}_i^{(j)}$$
$$= (f_i^{(j-1)} + d^{(j-1)}) - \tilde{f}_i^{(j-1)} \qquad \text{(from Eq. (1))}$$
$$= \sum_{l=0}^{j-2} d^{(l)} + d^{(j-1)} \qquad \text{(from the hypothesis } P(i, j-1))$$
$$= \sum_{l=0}^{j-1} d^{(l)}. \tag{8}$$

$$f_i^{(j)} - \tilde{f}_i^{(j)}$$
$$= (s_i^{(j)} + t_i^{(j)}) - (\tilde{s}_i^{(j)} + t_i^{(j)}) \qquad \text{(from Eq. (3))}$$
$$= s_i^{(j)} - \tilde{s}_i^{(j)}.$$

Next, the following holds from Eq. (2).

$$s_i^{(j)} - \tilde{s}_i^{(j)} = \max\{a_i^{(j)}, f_{i-1}^{(j)}\} - \max\{\tilde{a}_i^{(j)}, \tilde{f}_{i-1}^{(j)}\}. \tag{9}$$

We consider four cases in Eq. (9).

(i) When $a_i^{(j)} \geq f_{i-1}^{(j)}, \tilde{a}_i^{(j)} \geq \tilde{f}_{i-1}^{(j)}$:

$$\text{Eq. (9)} = a_i^{(j)} - \tilde{a}_i^{(j)} = \sum_{l=0}^{j-1} d^{(l)}. \qquad \text{(from Eq. (8))}$$

(ii) When $a_i^{(j)} \geq f_{i-1}^{(j)}, \tilde{a}_i^{(j)} < \tilde{f}_{i-1}^{(j)}$:

$$a_i^{(j)} \geq f_{i-1}^{(j)}$$
$$\Longleftrightarrow \tilde{a}_i^{(j)} + \sum_{l=0}^{j-1} d^{(l)} \geq s_{i-1}^{(j)} + t_{i-1}^{(j)} \quad \text{(from Eqs. (3) and (8))}$$
$$\Longleftrightarrow \tilde{a}_i^{(j)} + \sum_{l=0}^{j-1} d^{(l)} \geq \left(\tilde{s}_{i-1}^{(j)} + \sum_{l=0}^{j-1} d^{(l)}\right) + t_{i-1}^{(j)}$$
$$\qquad \text{(from the hypothesis } P(i-1, j))$$
$$\Longleftrightarrow \tilde{a}_i^{(j)} \geq \tilde{f}_{i-1}^{(j)} \qquad \text{(from Eq. (3))}$$

The last inequality contradicts the hypothesis in case (ii). Therefore, case (ii) is impossible.

(iii) When $a_i^{(j)} < f_{i-1}^{(j)}, \tilde{a}_i^{(j)} \geq \tilde{f}_{i-1}^{(j)}$: We can prove that case (iii) is impossible using the same method as used in case (ii).

(iv) When $a_i^{(j)} < f_{i-1}^{(j)}, \tilde{a}_i^{(j)} < \tilde{f}_{i-1}^{(j)}$:

$$\text{Eq. (9)} = f_{i-1}^{(j)} - \tilde{f}_{i-1}^{(j)} = \sum_{l=0}^{j-1} d^{(l)}. \qquad \text{(from the hypothesis } P(i-1, j))$$

From the four cases,

$$s_i^{(j)} - \tilde{s}_i^{(j)} = \sum_{l=0}^{j-1} d^{(l)}$$

holds. ∎

Theorem 1 The collision probability that considers delivery time is equal to the collision probability that does not consider delivery time.

Proof We assume that $(J_p, J_{p+1}, \ldots, J_q)$ is the waiting sequence in front of M_j, where $2 \leq p \leq q \leq n$. From the definition of the waiting sequence, $a_k^{(j)} < f_{p-1}^{(j)}$ holds for all $k = p, \ldots, q$. From Lemma 1, this inequality becomes $\tilde{a}_k^{(j)} + \sum_{l=0}^{j-1} d^{(l)} < \tilde{f}_{p-1}^{(j)} + \sum_{l=0}^{j-1} d^{(l)}$. When delivery time is not considered, $\tilde{a}_k^{(j)} = \tilde{f}_k^{(j-1)}$ holds. Therefore, this inequality becomes $\tilde{f}_k^{(j-1)} < \tilde{f}_{p-1}^{(j)}$. This inequality means that $(J_p, J_{p+1}, \ldots, J_q)$ is the waiting sequence in front of M_j in the case that delivery time is not considered (see Ref. [1]).

Therefore, the waiting sequence is the same as that found in the case when delivery time is not considered. Thus, a collision occurs at M_j in cases that consider delivery time only when a collision occurs at M_j in cases that do not consider delivery time. The converse of this is also true. ∎

5. Optimization Problems with Collision Probability

We first consider the tact time minimization problem with collision probability, which was presented in Ref. [2]. For the in-line machine model with delivery time, this problem is rewritten as follows.

Tact time minimization problem

Input: The number of jobs n, the number of machines m, the distributions of processing times, the number of buffers $b^{(j)}$, the delivery time $d^{(j)}$, and α $(0 \leq \alpha \leq 1)$ (specifying the collision probability).

Output: Tact time, such that the collision probability is less than or equal to α.

Objective function: $t_{\text{tact}} \longrightarrow \min$.

The constraint condition in this problem only relates to collision probability. Therefore, we obtain the following corollary from Theorem 1.

Corollary 1 The tact time minimization problem is equivalent to that which does not consider delivery time.

Next, we consider the buffer allocation problem with collision probability, which was presented in Ref. [1]. For the in-line machine model with delivery time, this problem is rewritten as follows.

Buffer allocation problem

Input: The number of jobs n, the number of machines m, the tact time t_{tact}, the distributions of processing times, the delivery time $d^{(j)}$, and α $(0 \leq \alpha \leq 1)$ (specifying the collision probability).

Output: The number of buffers $b^{(j)}$ in front of each machine, such that the collision probability is less than or equal to α.

Objective function: Total number of buffers $\sum_{j=1}^{m} b^{(j)} \longrightarrow \min$.

The constraint condition in this problem only relates to collision probability. Therefore, we obtain the following corollary from Theorem 1.

Corollary 2 The buffer allocation problem is equivalent to that which does not consider delivery time.

Theorem 1, Corollaries 1 and 2 are concrete examples which show the redundancy of delivery time in the in-line machine model.

6. Conclusions

In this paper, we considered the in-line machine model with delivery time. We proved that collision probability is the same as when delivery time is not considered. Therefore, when computing collision probability, delivery time is a redundant parameter.

Moreover, we considered the tact time minimization problem and the buffer allocation problem. We showed that these problems are equivalent to those which do not consider delivery time. Therefore, when considering these problems, delivery time is, again, a redundant parameter.

In Section 3, the variable $t_i^{(j)}$ was derived. The variable $t_i^{(j)}$ could be regarded as a constant. Therefore, without using random variables, we could easily show the result in Section 4. When analyzing similar stochastic cases to that in this paper via deterministic cases, our approach may be effective.

Acknowledgments The author would like to thank anonymous reviewers for their comments and suggestions from various viewpoints. The author also would like to thank Kengo Okamura, Akihito Niwa, and Sou Hukuoka for their discussions.

References

[1] Chiba, E.: Heuristics for the Buffer Allocation Problem with Collision Probability Using Computer Simulation, *Mathematical Problems in Engineering*, Vol.2015, Article ID 424370, 7 pages (2015).

[2] Chiba, E., Asano, T., Miura, T., Katoh, N. and Mitsuka, I.: Collision Probability in an In-Line Machines Model, *Trans. Computational Science XIII*, LNCS 6750, pp.1–12 (2011).

[3] Chiba, E., Fujiwara, H., Sekiguchi, Y. and Ibaraki, T.: Collision Probability in an In-Line Equipment Model under Erlang Distribution, *Trans. IEICE*, Vol.E96-D, No.3, pp.400–407 (2013).

[4] Hall, N.G. and Sriskandarajah, C.: A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process, *Operations Research*, Vol.44, No.3, pp.510–525 (1996).

[5] Heyman, D.P. and Sobel, M.J. (Eds.): *Stochastic Models: Handbooks in Operations Research and Management Science*, Vol.2, North-Holland, Amsterdam (1990).

[6] Kijima, M., Makimoto, N. and Shirakawa, H.: Stochastic Minimization of the Makespan in Flow Shops with Identical Machines and Buffers of Arbitrary Size, *Operations Research*, Vol.38, No.5, pp.924–928 (1990).

[7] Kleinrock, L.: *Queueing Systems, Volume I: Theory*, Wiley Interscience (1975).

[8] Kunisawa, K. and Honma, T. (Eds.): *Applied Queueing Dictionary*, Hirokawa Shoten (1971).

[9] Levner, E.M.: Optimal Planning of Parts Machining on a Number of Machines, *Automation and Remote Control*, Vol.12, pp.1972–1978 (1969).

[10] McCormick, S.T., Pinedo, M.L., Shenker, S. and Wolf, B.: Sequencing in an Assembly Line with Blocking to Minimize Cycle Time, *Operations Research*, Vol.37, No.6, pp.925–935 (1989).

[11] Pinedo, M.: Minimizing the Expected Makespan in Stochastic Flow Shops, *Operations Research*, Vol.30, No.1, pp.148–162 (1982).

[12] Pinedo, M.L.: *Scheduling: Theory, Algorithms, and Systems*, Springer, 4th edition (2012).

[13] Reddi, S.S. and Ramamoorthy, C.V.: On the Flow-Shop Sequencing Problem with No Wait in Process, *Operational Research Quarterly*, Vol.23, No.3, pp.323–331 (1972).

Eishi Chiba received a B.E. degree from Tohoku University in 2001, and an M.S. degree and a Ph.D. degree from the Japan Advanced Institute of Science and Technology in 2003 and 2006, respectively. He is currently a lecturer in the Department of Industrial & Systems Engineering at Hosei University, Japan. His research interests include operations research and their applications.