658.51

Calculation Methods of Maximum Waiting Space Required for a Process Involving Many Machine Groups*

By Genji YAMAZAKI**

A flow-shop has n machine groups in series, where the k -th machine group (machine group k) consists of m_k parallel servers, each having the same constant processing-time s_k . Jobs enter the first machine group and pass through the machine groups, in order, on a first-come-first-served queuing basis and on an unlimited queuing basis. Every job passes all machine groups. For arbitrary input of jobs, this paper shows that the maximum queue length and the maximum waiting time at the machine group k exist if $s_{k-1}/m_{k-1} \geq s_k/m_k$. The maximum queue length and waiting time are calculated and the waiting space required between machine groups is theoretically determined. The filter effect of the machine group is also explained.

1. Introduction and Summary

Flow-shop problems with constant processing-time at a machine group and with every job visiting every machine group are very common occurrences. An obvious example of such a flow-shop is an assembly line, where the workers or work stations represent the machines. Any group of machines served by a unidirectional, non-cyclic conveyor would be considered a flow-shop. A model is shown in Fig. 1, where m_k is the number of servers in machine group k and s_k is the known and constant processing-time at group k . The first machine must have a queue size which will accommodate all arriving jobs, but the allowable queue sizes in front of the other machine groups may be arbitrary. If a job is finished at machine group k and the queue in front of group $k+1$ is full, then the machine on which it has been processed cannot be used until the job can be put into the queue of group $k+1$ (so-called blocking state). If only the number in the system is of interest, there is no need to consider other than FCFS (first-come-first-served) discipline, since processing-times are constant. However, it is possible that the arrangement of machine groups could affect the waiting-time of a job, especially the time that could be spent at a machine group when the following machine group is full. Interest has centered on conditions under which the ordering of the machine groups will not affect the flow-time of a job (Avi-Itzhak⁽¹⁾, Friedman⁽²⁾, Suzuki and Kawashima⁽³⁾). The conditions under which it is known that a job's flow-time is independent of the machine group arrangement are:

- (a) Arbitrary number of machines per machine group and arbitrary allowable intermediate queue sizes (Suzuki and Kawashima⁽³⁾).

If we consider such systems where the order of performing services can be changed (for example, sequencing problems), (a) is one of effective measures. However, in a shop there occur more of cases in which processing order is fixed. Then, machine groups are arranged within limited space according to a given processing order. The largest problem in this case is a machine group which would become a bottle neck in a given system. Furthermore, when trouble may result in machine group k , a serious problem occurs on how to allocate waiting-space to each machine group in order to minimize influences on the other machine groups. The maximum queue length (or waiting-time) at each machine group would be one of good measures to analyze problems mentioned above. For arbitrary input process, Section 3 in this paper not only provides the criterion for existence of an upper bound of queue length (or waiting-time), but provides the upper bound of queue length (or waiting-time) at each machine group in Fig. 1. Furthermore, using theorems proved in Section 3, we deal with a few examples in Section 4.

Fig.1

2. Description of System

In this section, we give description of a system and notations in this paper. First, the machine group line (tandem queuing system) is considered as follows:

Model: n machine groups 1, 2, ---, n in tandem, where machine group k consists of m_k parallel machines, each having the same constant service-time (processing-time) s_k (see Fig. 1).

The Model is described by

$$\rightarrow (s_1 : m_1) \rightarrow (s_2 : m_2) \rightarrow \cdots \rightarrow (s_n : m_n) \rightarrow$$

* Received 19th January, 1974.

** Student of Doctor Course, Kogakuin Institute of Technology, Tokyo.

The following notations are used:

$$g_k = \text{G.C.M. } (m_1, m_2, \dots, m_{k+1}),$$

$$[b/a] = \text{maximum integer not exceeding } b/a,$$

$$((b/a)) = \text{minimum integer exceeding } b/a,$$

$$d_k = s_{k+1}/s_k$$

$$L_{g, \max}^{(k)} = \text{maximum waiting-time at the machine group } k,$$

$$W_{g, \max}^{(k)} = \text{maximum waiting-time at the machine group } k,$$

$$\lambda = \text{mean arrival rate (if it exists),}$$

$$\rho_k = \text{traffic intensity at the machine group } k$$

3. The Fundamental Results

For a given input process it is desirable that we explicitly analyze the distribution function of waiting-time at each machine group. However, at present, this is nearly impossible excepting certain special cases. For even if the machine group 1 can be analyzed with a given distribution function $G_1(\lambda)$ of interarrival-time to the machine group 1, distribution function $G_2(\lambda)$ of interarrival-time to the machine group 2 must be sought by the combination of service-time distribution and the number of machines in the machine group 1. Moreover, it is believed that, at present, there seems no effective method to obtain $G_2(\lambda)$ and it gradually becomes complicated as the input distribution function $G_3(\lambda)$ of the machine group 3 will have to be determined by combination of $G_2(\lambda)$, service-time distribution of the machine group 2 and the number of machines in the machine group 2. At present, results are given for a system where the input to the system is an exponential distribution with mean arrival rate λ . The service-time at each machine group is exponentially distributed and waiting-space in front of each machine group is unlimited. In this case, since it is proved that the input to each machine group is an exponential distribution with mean arrival rate λ , we can obtain separate analysis of individual machine groups. Another example is to get the analysis of the machine group 2. In this case, the following conditions are required: (1) The input to the machine group 1 is an exponential distribution. (2) The number of machines in the machine group are one. (3) The service-time distribution of the machine group 1 is arbitrary. (4) The service-time distribution of the machine group 2 is exponential. (5) The waiting-space in front of each machine group is unlimited. In the model shown in Fig. 1, in many applications the traffic intensity ρ_k often is designed close to unity. Then, we can consider that almost all machines at each machine group are busy. As a special case mentioned above we suppose that output sequence from the machine group 1 is as follows:

(a) There is a ware-house in front of the

machine group 1 and m_1 jobs at the machine group 1 just complete service in batch at any give epoch $t=0$. Continuously m_1 jobs at the machine group 1 just complete the service in batch at $t=s_1, 2s_1, \dots$.

For the output sequence of (a) from the machine group 1 (or the input sequence to the machine group 2), if an upper bound of queue length (or waiting-time) at the machine group k exists, for any given input sequence it is obvious that the queue length does not exceed the upper bound. Also, it can be expected that it will become a good upper bound when almost all machines are busy. The following theorems 1-5 are results for the model in Fig. 1.

Theorem 1. If $m_1 > m_2$ and $d_1 \leq 1/M_1$, where $d_1 = s_2/s_1$ and $M_1 = ((m_1/m_2))$, we have

$$L_{g, \max}^{(2)} \leq m_1 - m_2.$$

Proof. We deal with the following system:

$$\rightarrow (s_1 : m_1) \rightarrow (s_2 : m_2) \rightarrow.$$

Suppose that the machine group 2 is empty at epoch $t=0$ and a sequence of (a) is given as input sequence to the machine group 2 (which are called initial conditions). Let T denote time required for completing service m_1 jobs which arrive in batch at epoch $t=0$. First,

$$m_1 = a, m_2 + b, (\because m_1 > m_2) \quad \text{--- (1)}$$

where $a \geq 1, 0 \leq b < m_2; a, b = \text{integer}$.

(i) In the case of $b \neq 0$: From Eq. (1), m_1 jobs, arriving in batch at $t=0$, can separate the following total $a+1$ groups:

$$\left. \begin{array}{l} m_2 \\ m_2 \\ \vdots \\ m_2 \\ b, 1 \text{ group} \end{array} \right\} \begin{array}{l} a \text{ groups} \\ \\ \\ \\ 1 \end{array} \quad \left. \vphantom{\begin{array}{l} m_2 \\ m_2 \\ \vdots \\ m_2 \\ b, 1 \text{ group} \end{array}} \right\} (a+1) \text{ groups}$$

Then, m_2 jobs immediately receive service in batch from the machine group 2 at $t=0$ because of initial conditions. Subsequently, m_2 jobs will start to receive service in batch from the machine group 2 at $t=s_2, 2s_2, \dots, (a-1)s_2$. Also, the remaining b_1 jobs will start to receive service in batch from the machine group 2 at $t=a, s_2$ and will complete service at $t=(a+1)s_2$. Thus,

$$T = (a+1)s_2 \quad \text{--- (2)}$$

Here we define M as follows:

$$M_1 = ((m_1/m_2)) = ((a + b_1/m_2)) = a+1 \quad \text{--- (3)}$$

When $d_1 \leq 1/M_1$ holds, we get

$$d_1 = s_2/s_1 \leq 1/M_1 = 1/(a+1) \quad \text{--- (4)}$$

From Eq. (2) and inequality (4) we have

$$T = s_2 (a+1) \leq s_1 \quad \text{--- (5)}$$

Inequality (5) shows that m_1 jobs which arrive in batch at $t=0$ complete service until the next arrival epoch $t=s_1$ if inequality (4) holds. Therefore, all machines in the machine group 2 are idle at $t=s_1$. And m_1 jobs will arrive in batch from the machine group 1 at $t=s_1$, again. The same situation is repeated after $t=2s_1$. Then, right after each arrival epoch of m_1 jobs, m_2 jobs can receive service from the machine group 2 without waiting. Therefore, we have

$$L_{g, \max}^{(2)} = m_1 - m_2 \quad \text{---- (6)}$$

(ii) In the case of $b=0$. In this case m_1 jobs which arrive in batch at epoch $t=0$ can separate the following total a_1 groups,

$$\left. \begin{matrix} m_1 \\ m_2 \\ \vdots \\ m_2 \end{matrix} \right\} a_1 \text{ groups}$$

Hence, $T=a_1 s_2$, $M_1 = \lceil (m_1/m_2) \rceil = a_1$ ($\because b=0$). When $\alpha_1 \leq 1/M_1$ holds, we get

$$T = s_2 a_1 \leq s_1 \quad \text{---- (7)}$$

Therefore, in a similar way to the case (i), Eq. (6) also holds for the case of $b=0$. The proof is complete.

Theorem 1-1. If $m_1 > m_2$ and $\alpha_1 \leq 1/M_1$, we have

$$W_{g, \max}^{(2)} \leq \begin{cases} \lceil (m_1/m_2) \rceil - 1 \} s_2 & \text{for } m_1/m_2 = \text{integer} \\ \lceil (m_1/m_2) \rceil s_2 & \text{for } m_1/m_2 \neq \text{integer} \end{cases}$$

Proof. From $t=0$ we consider the same initial conditions as the proof of Theorem 1. (i) In the case of $m_1/m_2 = \text{integer}$: This case corresponds to $b=0$ in Eq. (1). Therefore, m_1 jobs which arrive in batch at $t=0$ can separate the total a_1 groups. The last group receives service from the machine group 2 with waiting-time $(a_1-1)s_2$ and has the longest waiting-time of m_1 jobs arriving at $t=0$. Also, $\lceil m_1/m_2 \rceil = a_1$. Thus, we have

$$W_{g, \max}^{(2)} = (a_1-1)s_2 = \{ \lceil m_1/m_2 \rceil - 1 \} s_2 \quad \text{---- (1.a)}$$

(ii) In the case of $m_1/m_2 \neq \text{integer}$: This case corresponds to $b \neq 0$ in Eq. (1). Therefore, m_1 jobs, arriving in batch at $t=0$, can separate the total a_1+1 groups. The last group receives service from the machine group 2 with waiting-time $a_1 \cdot s_2$ and has the longest waiting-time of m_1 jobs arriving at $t=0$. Also, $\lceil m_1/m_2 \rceil = a_1$. Thus, we have

$$W_{g, \max}^{(2)} = a_1 s_2 = \lceil m_1/m_2 \rceil s_2 \quad \text{---- (1.b)}$$

The proof is complete.

Theorem 2. No. queue ever occurs at the machine group 2 if $m_1 \leq m_2$ and $\alpha_1 \leq M_1'$ where $\alpha_1 = s_2/s_1$ and $M_1' = m_2/m_1$

Proof. We deal with the following system:

In this system, for the proof of theorem it is sufficient to show that $L_{g, \max}^{(2)} = 0$ for the following condition,

$$\alpha_1 = s_2/s_1 = M_1' \quad \text{or} \quad s_2 - s_1 M_1' = 0 \quad \text{---- (8)}$$

From $t=0$ we consider the same initial conditions as the proof of Theorem 1. Here the following notation is used in order to show the state of the machine group 2 at $t=j s_1$ ($j=0, 1, 2, \dots$).

$$t = j s_1 \rightarrow m_1 \quad m_j \cdot [t_j] \quad m_1 + m_j$$

where $\rightarrow m_1$: m_1 jobs arrive in batch from the machine group 1.

$m_j \cdot [t_j]$: m_j jobs, having the rest of remain at the machine group 2 at epoch $t=j s_1$.

$m_1 + m_j$: the total $m_1 + m_j$ jobs exist in the machine group 2 at epoch $t=j s_1$.

Then, we obtain the following table:

$t=0$	$\rightarrow m_1$	$0 \cdot [0]$	$m+0$
$=s_1$	$\rightarrow m_1$	$m_1 \cdot [s_1 - s_1]$	$m_1 + m_1$
$=2s_1$	$\rightarrow m_1$	$\left\{ \begin{matrix} m_1 \cdot [s_1 - 2s_1] \\ m_1 \cdot [s_1 - s_1] \end{matrix} \right\}$	$m_1 + 2m_1$

$= (M_1' - 1)s_1$	$\rightarrow m_1$	$\left\{ \begin{matrix} m_1 \cdot [s_1 - (M_1' - 1)s_1] \\ m_1 \cdot [s_1 - (M_1' - 2)s_1] \end{matrix} \right\}$	$m_1 + (M_1' - 1)m_1$
$= M_1' s_1$	$\rightarrow m_1$	$\left\{ \begin{matrix} m_1 \cdot [s_1 - s_1] \\ m_1 \cdot [s_1 - M_1' s_1] = m_1 \cdot [0] \\ m_1 \cdot [s_1 - (M_1' - 1)s_1] \end{matrix} \right\}$	$m_1 + (M_1' - 1)m_1$

where at $t=M_1' s_1$, $m_1 \cdot [s_1 - M_1' s_1] = m_1 \cdot [0]$ is obtained from Eq. (8). The conditions are quite the same at the machine group 2 in the case of $t=(M_1'-1)s_1$ and $t=M_1' s_1$. Hence, the same situation is repeated after $t=(M_1'+1)s_1$. Therefore, the maximum number of jobs in the machine group 2 become $m_1 + (M_1'-1)m_1 = M_1' m_1$. Then,

$$L_{g, \max}^{(2)} = M_1' m_1 - m_2 \quad \text{---- (9)}$$

where $M_1' = \lceil m_2/m_1 \rceil \leq m_2/m_1$ or $M_1' m_1 \leq m_2$ -- (10)

From Eq. (9) and inequality (10), we have

$$L_{g, \max}^{(2)} = 0 \quad \text{---- (11)}$$

This result agrees with Friedman's. The proof is complete.

Theorem 3. If $\alpha_1 \leq m_2/m_1$, we have $L_{g, \max}^{(2)} \leq m_1 - g_1$, where $g_1 = \text{G.C.M.}(m_1, m_2)$.

Proof. We deal with the following system: $(i) \rightarrow (s_1; m_1) \rightarrow (s_2; m_2) \rightarrow \dots$. In this system, from $t=0$ we consider the same initial conditions as the proof of Theorem 1. Since $g_1 = \text{G.C.M.}(m_1, m_2)$ we get $m_1 = g_1 m'_1$, $m_2 = g_1 m'_2$ (12), where m'_1 and m'_2 are mutually prime. Since the machine group 2 consists of m_2 parallel machines, each having the same constant service-time s_2 , we may divide system (i) into g_1 sets of the same subsystems, where the subsystem is described by $(ii) \rightarrow (s_1; m'_1) \rightarrow (s_2; m'_2) \rightarrow \dots$. From now on the system (ii) will be investigated. So long as there is no confusion, prime (') of m'_1 and m'_2 will be omitted. When $m_1 > m_2$, each machine of the machine group 2 can be numbered as in Fig. 2. Also, c indicates a job and each job can be numbered. Then, we obtain the following input sequence to the machine group 2:

Let iN_i denote the number of jobs to be served in the i -th machine whose jobs arrive at $t=js_1$ ($j=0, 1, 2, \dots$). Here we suppose that it offers service in the order of the machine's number according to the order of job's number. According to this service discipline, C_{m_1+i} , C_{2m_1+i} use the same machine (i -th machine) of the machine group 2 as C_i . Then, $iN_0 = ((m_1 - i)/m_2)$. A total $2m_1$ jobs arrive at the machine group 2 until right after $t=s_1$. The i -th machine has to serve $((2m_1 - i)/m_2)$ of $2m_1$ jobs. Hence, we get $iN_1 = ((2m_1 - i)/m_2) - iN_0 = ((2m_1 - i)/m_2) - ((m_1 - i)/m_2)$. In the same way, we obtain the following input sequence to the i -th machine:

$$iN_0 = ((m_1 - i)/m_2) \quad iN_1 = ((2m_1 - i)/m_2) - ((m_1 - i)/m_2) \quad \dots \quad iN_j = (((j+1)m_1 - i)/m_2) - ((jm_1 - i)/m_2) \quad (13)$$

For the proof of theorem, it is sufficient to show that $L_{g, \max}^{(2)} = m_1 - g_1$ for the following condition: $s_1/m_1 = s_2/m_2$ or $\alpha_1 = m_2/m_1$. When $s_1/m_1 = s_2/m_2$ holds, we obtain the following two results:

- (1) The 0-th machine is always busy. And, at the 0-th machine only one job arriving at $t=js_1$ ($j=0, m_1, 2m_1, \dots$) can receive service without waiting.
- (2) As the starting point at $t=js_1$, the i -th machine ($i=1, 2, \dots, m_1-1$) has completely the same conditions as the 0-th machine of (1), where if we assign a definite value to i , then a corresponding value of j_i ($=\text{integer}; 0 < j_i < m_2$) depends upon the previously selected value of i .

Proof of (1). From (13) we obtain the following input sequence to the 0-th

$$\text{machine:} \quad \begin{matrix} t=0 & t=s_1 & \dots & t=j s_1 & \dots \end{matrix} \quad \begin{matrix} {}_0N_0 & {}_0N_1 & \dots & {}_0N_{j_i} & \dots \end{matrix} \quad (14)$$

where ${}_0N_{j_i} = ((j_i+1)m_1/m_2) - (jm_1/m_2)$. Here the following notations are used:

$${}_0W_j(t) = \text{virtual waiting-time at epoch } t, \\ t = (js_1)^- : \text{immediately before epoch } t=js_1.$$

The total $\sum_{n=0}^{j-1} {}_0N_n$ jobs arrive at the 0-th machine in $[0, js_1)$. Since time required for $\sum_{n=0}^{j-1} {}_0N_n$ jobs completing service is $s_2 \sum_{n=0}^{j-1} {}_0N_n$, when $s_1 = s_2(m_1/m_2)$ (15), we have

$${}_0W_j(j s_1)^- = s_2 \sum_{n=0}^{j-1} {}_0N_n - j s_1 = ((j m_1 / m_2) s_2 - j s_1) = ((j m_1 / m_2) - j) s_2 > 0 \quad \text{for all } j; 0 < j < m_2 \quad (16)$$

$${}_0W_j(m_2 s_1)^- = ((m_2 m_1 / m_2) s_2 - (m_2 m_1 s_2) / m_2 = m_1 s_2 - m_1 s_2 = 0 \quad (17)$$

Eq. (16) and Eq. (17) show that the 0-th machine is always busy in $[0, m_2 s_1)$ and the virtual waiting-time at the 0-th machine becomes just 0 at $t=m_2 s_1$ (that is, one job of ${}_0N_{m_2}$ jobs arriving at the 0-th machine at $t=m_2 s_1$, can receive service without waiting). Let $L_0(j s_1)^-$ denote the number of jobs in the 0-th machine at $t=(j s_1)^-$. Then, from Eq. (16) and Eq. (17) we have

$$L_0(j s_1)^- = \sum_{n=0}^{j-1} {}_0N_n - [j s_1 / s_2] = ((j m_1 / m_2) - [j m_1 / m_2]) = 1 \\ \because b/a \neq \text{integer} \quad ((b/a)) = [b/a] + 1 \quad \text{for } 0 < j < m_2 \quad (18)$$

$$L_0(j s_1)^- = L_0(m_2 s_1)^- = 0 \quad \text{for } j=m_2 \quad \because b/a = \text{integer} \quad ((b/a)) = [b/a] \quad (19)$$

Hence, the input sequence to the 0-th machine in $[m_2 s_1, 2m_2 s_1]$ is identical with the sequence to the 0-th machine in $[0, m_2 s_1]$. The sequence is as follows:

$$t=m_2 s_1, \quad {}_0N_{m_2} = ((m_2+1)m_1/m_2) - ((m_2 m_1 / m_2)) = (m_1 / m_2) = {}_0N_0$$

$$t=(m_2+1)s_1, \quad {}_0N_{m_2+1} = ((m_2+2)m_1/m_2) - ((m_2+1)m_1/m_2) = ((2m_1/m_2) - (m_1/m_2)) = {}_0N_1$$

Therefore, we get again ${}_0W_j(2m_2 s_1)^- = 0$. In a similar way, we get ${}_0W_j(3m_2 s_1)^- = 0$, ${}_0W_j(4m_2 s_1)^- = 0$, The proof of (1) is complete.

Proof of (2). For $0 < j_i < m_2$, since m_1 and m_2 are mutually prime we have

$$j_i m_1 = a_i m_2 + b_i \quad 0 < b_i < m_2 \quad \text{--- (20)}$$

Also, if we use i for b_i we can rewrite Eq. (20) as follows:

$$j_i m_1 = a_i m_2 + i \quad \text{--- (20')}$$

where if we assign a definite value to i , then a corresponding value to j_i depends upon the previously selected of i because m_1 and m_2 are mutually prime. Using Eq. (20'), the input sequence to the i -th machine from $t=j_i s_1$ is identical with the one to the 0-th machine from $t=0$ as follows:

$$\begin{aligned} t=j_i s_1, \quad i N_{j_i} &= ((\{(j_i+1)m_1 - i\}/m_2)) - ((\{j_i m_1 - i\}/m_2)) = ((m_1/m_2)) = oN_0 \\ t=(j_i+1)s_1, \quad i N_{j_i+1} &= ((\{(j_i+2)m_1 - i\}/m_2)) - ((\{(j_i+1)m_1 - i\}/m_2)) = ((2m_1/m_2)) - ((m_1/m_2)) = oN_1 \quad \text{--- (21)} \end{aligned}$$

$t=(j_i+n)s_1, \quad i N_{j_i+n} = oN_n$
Therefore, if the i -th machine is idle at $t=j_i s_1$, we can apply the result of (1) for the case of i -th machine. Thus, we shall need only that the i -th machine be idle at $t=j_i s_1$. Let t' denote the last epoch at which the i -th machine is idle before $t=j_i s_1$. For $(j_i-1)s_1 < t' \leq j_i s_1$; $0 < j_i < j_i$ in the same manner of Eq. (16) and Eq. (17) we obtain

$$i W_g(j_i s_1)^- = s_2 \sum_{n=0}^{j_i-1} i N_n - (j_i - j_i) s_1 \quad \text{--- (22)}$$

Using (13), Eq. (22) and $s_1 = (m_1/m_2)s_2$, we get

$$i W_g(j_i s_1)^- = (j_i - i) s_2 / m_2 - ((\{j_i m_1 - i\}/m_2)) s_2 < 0 \quad \text{--- (23)}$$

For $t'=0$, we also obtain $i W_g(j_i s_1)^- = s_2 \sum_{n=0}^{j_i-1} i N_n - j_i s_1 = ((\{j_i m_1 - i\}/m_2)) s_2 - (j_i m_1 s_2 / m_1) = -i s_2 / m_2 < 0 \quad \text{--- (24)}$

Inequality (23) and inequality (24) show that the i -th machine becomes idle for sure at $t=j_i s_1$. Thus, using result (1), one of the jobs to arrive at the i -th machine at $t=(j_i+n)s_1$ ($n=0, 1, 2, \dots$) can receive service without waiting. The proof of (2) is complete.

From the results of (1) and (2) on page 1492 machine group 2 becomes always idle at epoch $[t=0, m_2 s_1, 2m_2 s_1, \dots]$, $[t=(m_2+1)s_1, (2m_2+1)s_1, \dots]$, $[t=(m_2+2)s_1, (2m_2+2)s_1, \dots]$, $\dots$, $[t=(m_2+m_1-1)s_1, (2m_2+m_1-1)s_1, \dots]$. In other words, there is always one job which can receive service from the machine group 2 without waiting at each arrival epoch. Thus, the maximum queue length in the subsystem (ii) becomes $L_{g, \max}^{(2)} = m'_1 - 1$. Therefore, the maximum queue length in the system (i), where the system (i) consists of g_1 sets of subsystems (ii), becomes

$$L_{g, \max}^{(2)} = g_1 L_{g, \max}^{(2)} = g_1 (m'_1 - 1) = g_1 m'_1 - g_1 = m_1 - g_1 \quad \text{--- (25)}$$

In the same manner, we can also obtain Eq. (25) for $m_1 \leq m_2$. The proof of theorem is complete.

Theorem 3.1. If $d_1 \leq m_2/m_1$ and $\max(m_2/m_1, m_1/m_2) \neq$ integer, we have

$$\begin{aligned} W_{g, \max}^{(2)} &\leq ((m'_1 - 1)/m'_1) s_1 \\ \text{where } m_1 &= g_1 m'_1 \text{ and } g_1 = \text{G.C.M.}(m_1, m_2). \end{aligned}$$

Proof. For $m_1/m_2 =$ integer ($m_1 > m_2$), $W_{g, \max}^{(2)}$ is given from Theorem 1.1. For $m_2/m_1 =$ integer ($m_1 \leq m_2$) $W_{g, \max}^{(2)}$ is given from Theorem 2. Hence we deal with the case of $\max(m_2/m_1, m_1/m_2) \neq$ integer here. For the proof of theorem, it is sufficient to show that $W_{g, \max}^{(2)} \leq ((m'_1 - 1)/m'_1) s_1$ for $d_1 = m_2/m_1$. When $m_1 > m_2$, from $t=0$ we consider the same initial conditions as the proof of Theorem 3 or Theorem 1. From the results of (1) and (2) on page 1492, it is obvious that we may consider only the 0-th machine in the machine group 2. At $t=(j s_1)^-$ the virtual waiting-time at the 0-th machine is

$$oW_g(j s_1)^- = ((j m_1 / m_2)) s_2 - (j m_1 / m_2) s_1 \quad \text{--- (16)}$$

(Note that prime (') of m'_1 and m'_1 is omitted here.) Let $m_1 = a_i m_2 + b_i$, $j b_i = a_j m_2 + b_j$, where $0 < b_i, b_j < m_2$. Substituting $j m_1$ and $j b_i$ into Eq. (16), we obtain

$$oW_g(j s_1)^- = (m_2 - b_j) s_2 / m_2 < s_2 \quad \text{--- (26)}$$

From (14) we have $oN_j = ((\{j+1\} m_1 / m_2)) - ((j m_1 / m_2)) = a_i + ((\{b_i + b_j\} / m_2)) - ((b_j / m_2))$
 $\therefore oN_j = \begin{cases} a_i & \text{for } b_i + b_j \leq m_2 \\ a_i + 1 & \text{for } b_i + b_j > m_2 \end{cases} \quad \text{--- (27)}$

where b_i is constant and b_j depends upon j . From inequality (26), $oW_g(j s_1)^-$ is larger when b_j is smaller, but oW_g is less than s_2 . From Eq. (27) the arrival number of jobs at $t=j s_1$ for $b_j > m_2 - b_i$ increases by one in comparison with the arrival number of jobs in the case of $b_j \leq m_2 - b_i$. After all, one of the jobs arriving at $t=j s_1$ has the longest waiting-time where b_j is the minimum integer satisfying $b_j > m_2 - b_i$, that is, $b_j = m_2 - b_i + 1$. Let us denote the longest waiting-time by $oW_{g, \max}^{(2)}$. Then,

$$oW_{g, \max}^{(2)} = (m_2 - b_i) s_2 / m_2 + a_i s_2 = \{a_i + (b_i - 1) / m_2\} s_2 = (m'_1 - 1) s_1 / m_1 \quad \text{--- (28)}$$

In the same manner, we can also obtain Eq. (28) for $m_1 \leq m_2$, which completes the proof.

The following theorem is self-evident.

Theorem 4. $\rightarrow (s_{k-1} : m_{k-1}) \rightarrow (s_k : m_k) \rightarrow$ In this system, if no queue ever occurs in the machine group k , the output sequence from the machine group k is identical with the one from the machine group $k-1$. Theorems 1, 2, 3 are applicable to relations between the machine group $k-1$ and machine group k . And, actually, it will be safe if we can determine waiting-space with an estimation of $L_{g, \max}^{(k)}$ obtained from Theorems 1-4. But in case of successive servicing of productive process at a shop, the servicing capacity of each machine group is practically the same (This situation can be regarded in Fig. 1 as $s_k/m_k = \text{const.}$).

Therefore, if we apply Theorems 1-3 to the machine group $k-1$ and the machine group k ($k \geq 3$), it will be overestimated, because Theorems 1-3 consider $L_{g, \max}^{(k)}$ in the machine group k with a large number of jobs waiting in front of the machine group $k-1$. In the model in Fig. 1, even if we consider the output sequence of (a) from the machine group 1 on page 1490 with the warehouse in front of the machine group 1, the machine groups 2- $k-2$ together with the machine group $k-1$ have a filter-effect on the machine group k . When we consider this filter-effect, we get the following theorem:

Theorem 5. In the model in Fig. 1, if $s_k/m_k = \text{const.}$ holds and all machines at every machine group are always busy, we have $L_{g, \max}^{(k)} \leq g_{k-2} - g_{k-1}$.

Proof. For $m_1 > m_2$, we consider the subsystem (ii) of the proof of Theorem 3. From $t=0$, we also consider the same initial conditions as the proof of Theorem 3. Using Eq. (16), for the i -th machine in the machine group 2 we have

$$\lambda W_g^{(2)}(js_1) = \lambda W_g^{(2)}\{(j-i)s_1\} = (1 - b_i/m_1')s_2 \quad j \geq m_1' - 1; 0 \leq b_i < m_1' \quad \text{--- (29)}$$

where b_i depends upon i . Here we shift the original point of time from $t=0$ to $t=js_1$. Let us denote a new axis of time obtained above by t' . From Eq. (29), the jobs being served at $t'=0$ ($t=js_1$) is completed one after another at the following epoch:

$$t' = s_2/m_1', 2s_2/m_1', \dots, m_1's_2/m_1'.$$

Jobs arriving after $t'=0$ also will complete service one after another at the following epoch:

$$t' = s_2/m_1' + s_2, 2s_2/m_1' + s_2, \dots, m_1's_2/m_1' + s_2, s_2/m_1' + 2s_2, \dots$$

because for $j \geq m_1' - 1$ each machine is always busy from the results of (1) and (2) on page 1492. Therefore, after $t' = (1/m_1')s_2$ the output sequence from the machine group 2 in the subsystem (ii) becomes a constant interdeparture time of s_2/m_1' . Since the original system (i) consists of g_1 sets of subsystems (ii), after all, g_1 jobs leave in batch from the machine group 2 in the system (i) with constant interdeparture time of s_2/m_1' . Thus, from the stand point of the machine group 3, for a given output sequence of (a) from the machine group 1 if $s_1/m_1 = s_2/m_2$ holds, warehouse $\rightarrow (s_1:m_1) \rightarrow (s_2:m_2) \rightarrow \text{reduces to warehouse} \rightarrow (s_2/m_1':g_1) \rightarrow$. Using Theorem 3, we get

$$L_{g, \max}^{(3)} = g_1 - G.C.M. (g_1, m_3) = g_1 - g_2 \quad \text{--- (30)}$$

In the same way of $L_{g, \max}^{(3)}$, for $L_{g, \max}^{(k)}$ ($k \geq 3$) we get

$$L_{g, \max}^{(k)} = G.C.M. (m_1, m_2, \dots, m_{k-1}) - G.C.M. (m_1, m_2, \dots, m_k) = g_{k-2} - g_{k-1} \quad \text{--- (31)}$$

Similarly, for $m_1 \leq m_2$ we obtain the same result, which completes the proof.

When we design the model in Fig. 1, Theorems 1-5 are one of effective measures for discussion on the following problems: which the machine group becomes a bottle neck in the system? How to allocate the waiting-space to each machine group in order to minimize influences on other machine groups? Moreover, it can be useful methods for jobs to find out the longest waiting-time.

4. Examples

Example 1. (1) $\xrightarrow{G(\lambda)} (5:10) \rightarrow (3:6) \rightarrow$
(2) $\xrightarrow{G(\lambda)} (3:6) \rightarrow (5:10) \rightarrow$ (minute)

In the system (1) the input distribution of the machine group 1 is uncertain but the mean arrival rate λ is approximately equal to 1/0.55 (1/min.). Determine the minimum waiting-space in front of the machine group 2 in the system (1) required for no-blocking. If the order of performing service can be changed (see system (2)), what will become of its waiting space?

Solution. In the system (1), we have $\rho_1 = \lambda s_1/m_1 = 10/11$, $\rho_2 = \lambda s_2/m_2 = 10/11$, $s_1/m_1 = s_2/m_2$. Namely, $d_1 = s_2/s_1 = m_2/m_1$ holds and ρ_1 and ρ_2 are close to unity. Using Theorem 3, we have $L_{g, \max}^{(2)} = m_1 - g_1 = 10 - 2 = 8$. Therefore, the waiting-space in front of the machine group 2 is sufficient with space where 8 units (8 jobs) can be permitted to wait. Next, in the system (2), $d_1 = s_2/s_1 = m_2/m_1$ holds (note that $m_1=6$, $m_2=10$). Using Theorem 3, we have $L_{g, \max}^{(2)} = m_1 - g_1 = 6 - 2 = 4$. Therefore, the waiting-space in front of the machine group 2 is sufficient with space where 4 units can be permitted to wait, i.e. the waiting-space in front of the machine group 2 in the system (2) reduces to half of the waiting-space in front of the machine group 2 in the system (1).

Example 2. In the system (1), we want to make the waiting-space in front of the machine group 2 as small as possible. We carry out this work effectively by shortening the service-time of the machine group 2. How much can the service-time be shortened?

Solution. Since $m_1 > m_2$ and $M_1 = (m_1/m_2) = (10/6) = 2$, we obtain the following relations:

$$s_1^*/s_1 \leq 1/M_1 \quad \therefore s_1^*/5 \leq 1/2 \quad \text{or} \quad s_1^* \leq 2.5 \quad (\text{min.})$$

Accordingly, when $s_1^* \leq 2.5$ (min.) we have from Theorem 1

$$L_{g, \max}^{(2)} \leq m_1 - m_2 = 10 - 6 = 4.$$

By shortening to time 0.5 (min.), (as to service-time of the machine group 2) the waiting-space in front of the machine group 2 in the system (1) can be reduced to half.

Example 3. In a shop, raw materials can be worked into manufactured goods by the following process:

m.g. 1 m.g. 2 m.g. 3 m.g. 4 m.g. 5
 $\rightarrow (4:3) \rightarrow (6:12) \rightarrow (2:4) \rightarrow (4:8) \rightarrow (5:10) \rightarrow$

When the raw materials are always supplied sufficiently, determine the minimum waiting space in front of each machine group required for successive servicing.

Solution. $s_k/m_k = 1/2$

(constant; $k=1, 2, \dots, 5$)

machine group 2: Using Theorem 3, we have

$$L_{g,max}^{(2)} = m_1 - g_1 = 8 - 4 = 4.$$

Therefore, the waiting-space in front of the machine group 2 is sufficient with 4-unit space (that is, space where 4 jobs can be permitted to wait.)

machine group 3: Using Theorem 3, we have

$$L_{g,max}^{(3)} = m_2 - G.C.M. (m_2, m_3) = 12 - 4 = 8.$$

Therefore, the waiting-space in front of the machine group 3 is sufficient with 8-unit space.

machine group 4: Since $s_4/s_3 = [m_4/m_3] = 2$ holds, no queue ever occurs at the machine group 4 (from Theorem 2).

machine group 5: Since no queue ever occurs at the machine group 4, we can apply Theorem 4 to this case. Using Theorem 4, we can calculate the waiting-space in front of the machine group 5 from Theorem 3 by the relationship between the machine group 3 and machine group 5. From Theorem 3, we have

$$L_{g,max}^{(5)} = m_3 - G.C.M. (m_3, m_5) = 4 - 2 = 2.$$

Therefore the waiting-space in front of the machine group 5 is sufficient with 2-unit space.

Example 4. Disregarding the initial conditions in the system of Example 3, we want to make the waiting-space at each machine group as small as possible. To what extent should the waiting-space at each machine group be allowed?

Solution. Since we disregard the initial conditions and consider the successive servicing, we can apply Theorem 5 to this case. From Theorem 5, we have

$$g_0 = 8, g_1 = 4, g_2 = 4, g_3 = 4, g_4 = 2$$

$$L_{g,max}^{(2)} = g_0 - g_1 = 4, \quad L_{g,max}^{(3)} = g_1 - g_2 = 0,$$

$$L_{g,max}^{(4)} = g_2 - g_3 = 0, \quad L_{g,max}^{(5)} = g_3 - g_4 = 2.$$

Comparing with the results of Example 3, $L_{g,max}^{(3)}$ decreases from 8-unit to 0 unit. Thus, we can reduce the waiting-space in front of the machine group 3 (8 $\sim$ 0).

Acknowledgement

The author would like to acknowledge the continuing guidance and encouragement of Professor Raiji Kato.

References

- (1) Avi-Itzhak, B., Management Sci., 11-5 (1965-3), 565.
- (2) Friedman, H. D., Operations Research, 13-1 (1965-1), 121.
- (3) Suzuki, T. and Kawashima, T., J. Operations Res. Soc. Japan, 17-3 (1974), 133
- (4) Suzuki, T., J. Operations Res. Soc. Japan, 5-4 (1963), 149.
- (5) Conway, R. W., et al, Theory of Scheduling, (1967) Addison-Wesley.