ESTIMATING THE THROUGHPUT OF A CYCLIC ASSEMBLY SYSTEM

IZAK DUENYAS
Department of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 92-49

September 1992
Revised April 1993

ESTIMATING THE THROUGHPUT OF A CYCLIC ASSEMBLY SYSTEM

IZAK DUENYAS

Department of Industrial and Operations Engineering

The University of Michigan

Ann Arbor, Michigan 48109

Abstract

We consider a production system consisting of several fabrication lines feeding an assembly station. The machines in the fabrication lines and at assembly are assumed to have general processing time distributions. Releases to the system are governed by the CONWIP protocol. We model this system as an assembly-like queue and develop approximations for the throughput of the system. Comparisons with simulations show that this approximation is robust over a wide range of conditions. Finally, we observe that throughput tends to be higher when machines with higher mean processing times and/or higher variances are in fabrication rather than assembly.

1 Introduction

Assembly-like queues arise in a variety of practical situations. They are especially prevalent in manufacturing systems. A typical example in electronics manufacturing is the production of multi-plane circuit boards (PCB's) which are manufactured by fabricating the layers separately and then laminating them together. In fact, any production system where several sub-assemblies are produced separately, and then assembled together will give rise to assembly-like queues.

Despite the prevalence of assembly-like queues in many manufacturing environments, little work has been done on these queues due to their analytical intractability. The majority of queueing network models do not handle assemblies. The bulk of the queueing models that handle assemblies represent the fabrication lines feeding assembly as single machines, an assumption that limits their applicability (Ammar (1980), Bhat (1986), Bonomi (1987), Hopp and Simon (1989), and Lipper and Sengupta (1986)). Gershwin (1991) and Mascolo et al. (1991) have focused on assembly systems with acyclic or tree structured networks and their models allow for multiple machines including many assembly/disassembly machines. However, they only treat the case where all the machines are assumed to have the same processing time distribution. Baker et al. (1990, 1993) have used simulation to address the problem of allocating work in assembly systems. Other work has concentrated on obtaining structural results such as monotonicity and concavity results for these networks (e.g., Adan and Van der Wal (1989), Ammar and Gershwin (1989), Bacelli et al. (1989)). In a recent and very comprehensive survey of manufacturing flow line systems, Dallery and Gershwin (1992) noted the lack of approximations for assembly systems and emphasized the need for more work in this area.

An issue that complicates the modelling of assembly systems is that assembly-like queues are unstable unless some feedback mechanism is used to link releases to outputs (Harrison (1973)). In many manufacturing systems, the method for controlling releases is MRP, in which releases are scheduled by subtracting fixed lead times from due dates. More recently, pull systems, such as kanban, have become popular due to the success of Japanese just-in-time methods. (Monden(1983)).Despite the great interest in pull systems in the last decade, assembly systems under pull release mechanisms have received very little interest. Duenyas and Hopp (1992a, 1992b) focus on assembly systems under the CONWIP (Constant Work-In-Process) release mechanism. Duenyas and Hopp (1992a) assume that all machines have exponential processing time distributions, while Duenyas and Hopp (1992b) assume that the machines have deterministic processing times and are subject to random failures. The

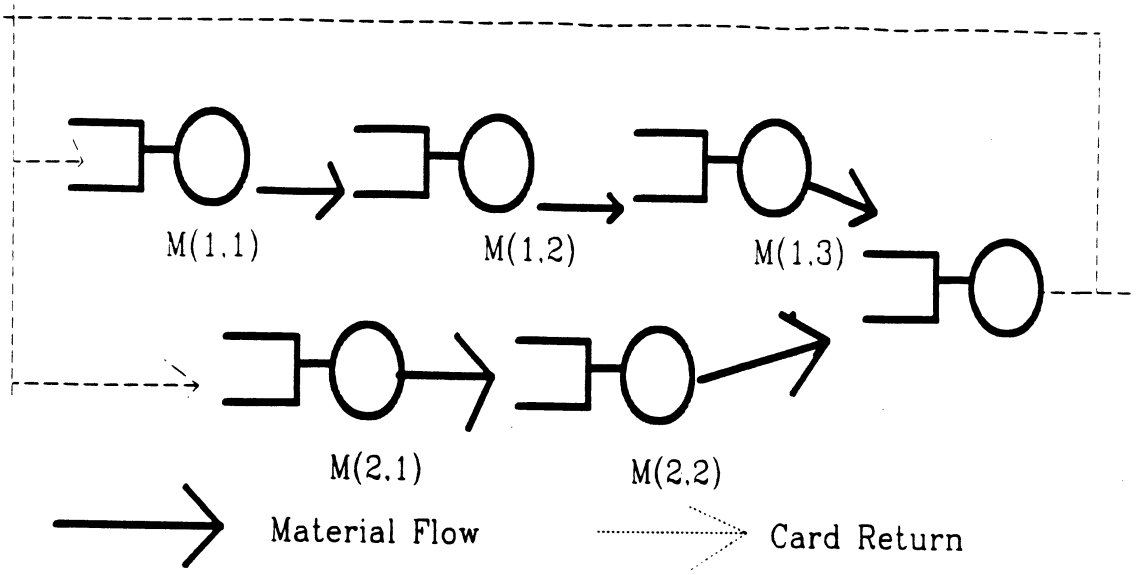

Figure 1: CONWIP Assembly System

time between failures as well as the time between repairs is assumed to be exponentially distributed.

Assembly systems operating under kanban can be modelled using Markov models if all the machines are assumed to have exponential processing time distributions. However, this requires a very large state space for any realistic system. Also, in most manufacturing systems, the processing times are less variable than the exponential. For these reasons, in this paper, we will restrict our attention to the CONWIP (CONstant Work-In-Process) production control system. CONWIP is a broadly applicable pull system that offers many of the benefits of kanban (e.g., WIP control), but is simpler to model than kanban. Spearman et al. (1989, 1990) discuss the advantages of CONWIP, while Spearman and Zazanis (1992) show that a tandem CONWIP system always has higher throughput than the equivalent kanban system when both systems have the same number of cards in the system. These advantages of CONWIP, combined with the relative simplicity of modeling it makes it an attractive pull mechanism.

In a CONWIP system, the total amount of work is held constant by authorizing production of a new unit only when an output occurs. Release mechanisms based on keeping

the total amount of work in the system constant are very common in industry, especially in wafer fabrication (Ehteshami et al., 1992). The regulation of the amount of work in a CONWIP system can be accomplished with cards as in the kanban system. Each time a job is completed, its card is removed, and sent to the front of the line to authorize the start of a new job. We note that once production is authorized at the first machine, jobs are pushed between machines elsewhere in the line. In an assembly system operating under CONWIP (displayed in Figure 1), jobs exit only from the assembly. Hence, whenever a job is finished at the assembly machine, a card is sent to the first machine in each fabrication line authorizing the start of work on a new unit. However, note that these simultaneously released jobs are not necessarily for the same assembly unless all the fabrication lines have the same card counts (WIP levels).

In this paper, we develop an approximation for throughput, and average WIP levels at each machine, for a CONWIP assembly system where all machines are assumed to have general processing time distributions. The approximation in this paper generalizes the approach in Duenyas and Hopp (1992a) where a throughput approximation was derived under the assumption that processing times are exponential. The approximation in this paper can be used along with simulation in the configuration of fabrication/assembly lines. In particular, the approximation can be used along with the procedures developed in Duenyas and Hopp (1992a) for the problems of setting WIP levels and capacity allocation. Therefore, our approximation can be used by practitioners implementing a CONWIP release mechanism for design and analysis of their assembly systems.

The remainder of this paper is organized as follows. In the next section, we introduce our notation and problem formulation. In Section 3, we develop an upper-bound for throughput and compute this upper bound approximately by making use of an approximation for the throughput of closed queueing networks with general processing times by Shantikumar and Gocmen (1983). In Section 4, we make use of this upper bound to develop an approximation for the throughput of the assembly system. In Section 5, we conduct a simulation study to

test our approximation against simulation and show that the approximation seems to be robust for a broad range of cases. Section 6 concludes the paper.

2 Problem Formulation

We consider k production lines with m_j machines and n_j jobs at each line j as shown in Figure 1. At each line j, jobs start at machine $(j,1)$ and after being processed at machine (j,i) move to machine $(j,i+1)$. We assume that successive processing times at machine (j,i) are independent with finite mean, $x_{j,i}$ and variance $\sigma^2_{j,i}$.

After completing work at machine (j,m_j), a job joins the assembly queue. If there is at least one job from each line in the assembly queue, then the assembly operation begins. Assembly times are independent and have finite mean, m_A and variance σ^2_A. When the processing at assembly is finished, an output occurs and this sends a signal to machines $(j,1), j = 1,\ldots,k$ to add a new job to their queues. We define N_t as the number of outputs until time t, starting from time 0. We are interested in finding the throughput for the system, $\theta = lim_{t\to\infty} N_t/t$.

In the special case where each line has exactly one job (i.e., $n_j = 1$ for each line), it is possible (but, very tedious) to obtain an exact expression for the average cycle time and hence for the throughput, given the distribution of processing times at each machine, since the times between outputs are i.i.d. in this case. However, for values of n_j greater than 1, neither the interoutput times nor the cycle times are i.i.d.. Hence, we develop an approximation for throughput, which we describe in the next two sections.

3 An Upper Bound for Throughput

In this section, we develop an upper bound for the throughput of our assembly system. We let $F_{j,i}$ denote the "information" that we have on the processing time at machine (j,i). For example, if we know the distribution of the processing time, then $F_{j,i}$ denotes the c.d.f. of

the distribution. However, in our approximation, we will only need the first two moments of the processing time. Hence, if we only have that information, we can let $F_{j,i}$ be an array that consists of the mean and variance of the processing time at machine (j,i) (i.e., $F_{j,i} = [x_{j,i}, \sigma^2_{j,i}]$). We let F be a two-dimensional array containing the $F_{j,i}$ values. We also denote the processing time "information" for the assembly operation by F_A. The work-in-process inventory in fabrication line j is n_j, and n is an array containing the n_j values. We let $\{F/F_A/n\}$denote the assembly system shown in Figure 1. Let $\theta\{F/F_A/n\}$ denote the steady-state average throughput of $\{F/F_A/n\}$.

To obtain an upper bound on the throughput of the assembly system, we compare the assembly system in Figure 1 to several tandem closed queueing networks. In particular, let $\{F_r/F_A/n_r\}$ denote a closed tandem queueing network that consists of machines $(r,1),\ldots,(r,m_r)$ in sequence with the assembly machine at the end (where the assembly machine no longer "assembles", but instead, processes single jobs). In this case, jobs start at machine $(r,1)$, move to $(r,i+1)$ after (r,i) and to the assembly machine after (r,m_r). After completing work at the assembly, jobs return to machine $(r,1)$. Let $\theta\{F_r/F_A/n_r\}$ denote the throughput of this system. Then, we have the following

Proposition 1 *(Duenyas and Hopp (1992a))*

$$\theta\{F/F_A/n\} \leq \min_r \theta\{F_r/F_A/n_r\}. \tag{3.1}$$

The above proposition provides an upper bound for the throughput of the assembly system. Using Proposition 1, the throughput of an assembly system with k fabrication lines is bounded above by the minimum of the throughput of k closed queueing networks formed from the assembly system. However, we note that for closed queueing networks with general processing times, exact results for throughput are not available although very effective approximations have been developed (e.g., Whitt(1984) and Shantikumar and Gocmen (1983)). Therefore, we can only compute an approximation for this upper bound. In calculating approximations for an upper bound on throughput and for throughput, we make use

of the approximation procedure developed by Shantikumar and Gocmen for estimating the throughput of a closed queuing network. This approximation requires only the knowledge of the mean and the variance of the processing time at each station. Hence, for our assembly system, we will also require only the information on mean and variance of processing times at each station. The approach of Shantikumar and Gocmen is based on decomposing the closed queueing network. Their approach can be summarized as follows:

Procedure for estimating the throughput of a closed queueing network (Shantikumar and Gocmen (1983))

1. Obtain an initial approximation for the server utilizations by replacing the general processing times by exponential processing times.

2. Evaluate approximations for the arrival rate and the squared coefficient of variation of the arrival process to each machine.

3. Treat each server in isolation. Using the approximation for the arrival process, obtained in step 2 for the arrival process, obtain an approximate probability distribution for the number of jobs in an equivalent single server (GI/G/1) queue. Find the load-dependent service rates $\{\mu_n\}$ for the single server queue $(M/M(n)/1)$ with Poisson input and exponential service times that has the same arrival rate, mean service time, and the same probability distribution of the number of jobs in the system as that of the $GI/G/1$ queue.

4. Obtain a new approximation for the server utilizations of the servers in the network with the general service times replaced by exponential service times with load dependent service rates obtained in Step 3. If the new values for server utilizations are sufficiently close to the previous ones, then stop, else go to step 2.

We refer the reader to Shantikumar and Gocmen (1983) for further details on the throughput approximation for a closed queueing network. We can use the above approximation for closed queueing networks along with Proposition 1 to obtain an approximate upper

bound on the throughput of the assembly system. However, we note that Shantikumar and Gocmen point out that their approximation is not guaranteed to converge if the coefficient of variation of the processing times is greater than 1. However, as we have previously noted, in most manufacturing systems, processing times have distributions that are much less variable than the exponential. Therefore, assuming that the coefficient of variation of the processing time is less than 1 is not a restrictive assumption in most practical situations. Since, in developing our approximation for the throughput of the assembly system, we will make use of the approximation by Shantikumar and Gocmen for closed queueing networks, we will also restrict our attention to systems with machines that have coefficient of variation of processing time less than 1. We next use the upper bound developed in this section in a procedure for estimating the throughput of the assembly system.

4 An Approximation for Throughput

Our approximation procedure is based on estimating the delay that a job from a particular line experiences waiting for jobs from other lines before its assembly can begin. A job in a certain line r is delayed at assembly if, when it is that particular job's turn to be served by the assembly machine, there is not at least one job from all other lines. When this occurs, the job in line r has to wait for the other jobs to arrive. Hence, to develop an approximation for the throughput, we first develop an approximation for this delay. We start by developing our approximation for a system with two lines, and then generalize our approach to more than two lines.

Consider an assembly system with only two lines. We let W_1 be the amount of time that a job from line 1 has to wait at assembly for a job from line 2. To calculate EW_1, we condition on the position of the jobs in line 2 at the time that the job from line 1 arrives at assembly. Let N_i be the number of jobs in line 2, machine i. Also, denote the number of jobs from line 2 waiting in front of assembly as N_{m_2+1}. Obviously, if $N_{m_2+1} > 0$, then there is at least one job from line 2 already waiting at assembly, and hence the expected

delay that a job from line 1 experiences waiting for a job from line 2 at assembly is 0 in this case. In general, we have

$$EW_1 = \sum_{i=1}^{m_2+1} E[W_1|N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0]P(N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0) \tag{4.2}$$

$$EW_1^2 = \sum_{i=1}^{m_2+1} E[W_1^2|N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0]P(N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0) \tag{4.3}$$

and

$$Var[W_1] = EW_1^2 - (EW_1)^2 \tag{4.4}$$

Calculating the conditional expectations in (4.2) and (4.3) requires that we also condition on the amount of processing that the job in line 2 "closest to" assembly has had at the machine that it is being processed at the time the job from line 1 arrives at assembly. However, since our aim is to get a rough estimate of the expected waiting time, we ignore the amount of processing that the job may already have had and approximate the conditional expectation by

$$E[W_1|N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0] \simeq \sum_{p=i}^{m_2} x_{2,p} \tag{4.5}$$

Similarly, we let

$$E[W_1^2|N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0] \simeq \sum_{p=i}^{m_2} \sigma_{2,p}^2 + (\sum_{p=i}^{m_2} x_{2,p})^2 \tag{4.6}$$

Approximating the probabilities in (4.2) is more difficult however, since we do not know the distribution of the jobs in the network. Hence, we approximate these by supposing that while jobs in line 1 have to wait for jobs in line 2 for their assembly operation, jobs in line 2 are independent of jobs in line 1 and start their assembly operation regardless of whether or not there are jobs from line 1 in assembly. This makes line 2 a regular closed queueing network, and we can use the approximation by Shantikumar and Gocmen to calculate utilizations for line 2. Furthermore, the approximation procedure by Shantikumar and Gocmen approximates the closed queueing network by a load-dependent exponential queueing network, and calculates Buzen's coefficients (Buzen (1983)) for this network at each iteration. Let $G(i,j), i = 0,\ldots,n_2, j = 1,\ldots,m_2+1$ denote the Buzen's coefficients

calculated by the Shantikumar and Gocmen approximation procedure on its last iteration for the closed queueing network formed by line 2 and the assembly machine. Then, we have

$$P[N_i > 0, \sum_{p=i+1}^{m_2+1} N_p = 0] \simeq \frac{G(n_2, i) - G(n_2, i-1)}{G(n_2, m_2+1)} \tag{4.7}$$

Hence, we can estimate the first and second moments of the expected delay that jobs from line 1 experience waiting for jobs from line 2 by using (4.2), (4.3), (4.4) along with (4.5), (4.6), and (4.7). Given that we have obtained an estimate of the delay that jobs from line 1 experience waiting for jobs from line 2, one obvious way to obtain an approximation for the throughput would be to treat line 1 itself as a closed queueing network. However, since jobs in line 1 experience a delay at assembly, we would change the assembly machine's mean processing time to $x_A^{'}$ and its variance to $\sigma_A^{2'}$, where

$$x_A^{'} = x_A + EW_1 \tag{4.8}$$

and

$$\sigma_A^{2'} = \sigma_A^2 + Var[W_1] \tag{4.9}$$

Then, we would have a new closed queueing network formed by all the machines from line 1 along with the "revised" assembly machine. We describe this new network by $\{F_1/F_A + W_1/n_1\}$ where $F_A + W_1$ denotes that the mean and variance of delay experienced by jobs from line 1 waiting for jobs from line 2 at assembly has been added, respectively, to the mean and variance of the processing time at assembly. We could then use the Shantikumar and Gocmen approximation for closed queueing networks once again to obtain an approximation for the throughput. Notice however, that just as jobs from line 1 wait for jobs from line 2 at assembly, the reverse is true as well. Hence, to capture the effect that both lines have on each other, we also need to calculate EW_2. In fact, we propose starting with $\{F_1/F_A/n_1\}$ to calculate EW_2, and $Var[W_2]$, then using $\{F_2/F_A + W_2/n_2\}$ to calculate EW_1, and $Var[W_1]$ and continuing in this manner until the throughput converges. The only remaining issue that we have to resolve is the choice of line 1 and line 2. We let line 1 be the line that sets the upper bound on throughput. That is, we let $h = argmin_r \theta\{F_r/F_A/n_r\}$. We renumber

line h as line 1 since that line actually is "closest" to the throughput of the network. We can now present our procedure for computing the throughput estimate θ_{ap} for the assembly system.

Procedure for Computing θ_{ap} (2 lines)

1. For $i = 1,2$ compute $\theta\{F_i/F_A/n_i\}$, using the throughput approximation for closed queueing networks in Shantikumar and Gocmen. Let $h = argmin_i\theta\{F_i/F_A/n_i\}$. Renumber line h as line 1. Let $\theta_1 = \min_i \theta_i\{F_i/F_A/n_i\}$. Let $EW_1 = 0$, and $VarW_1 = 0$.

2. Compute EW_2, and $Var[W_2]$ using (4.2) and (4.4), and $\{F_1/F_A + W_1/n_1\}$.

3. Compute EW_1, and $Var[W_1]$ using (4.2) and (4.4), and $\{F_2/F_A + W_2/n_2\}$.

4. $\theta_{ap} = \theta\{F_1/F_A + W_1/n_1\}$. If $|\theta_{ap} - \theta_1| < \epsilon$ for a prespecified ϵ, then stop. Else, let $\theta_1 = \theta_{ap}$, go to 2.

We note that the above procedure uses the approximation in Shantikumar and Gocmen for closed queueing networks twice in each iteration. Since that procedure provides only approximate values for the utilization of machines, and the throughput of the closed queueing network, and not the actual values, we can not guarantee that the above procedure will converge. However, in the large number of experiments that we performed, we could not find a case where the above procedure did not converge. In fact, convergence was so quick that using $\epsilon = 10^{-3}$, in all our test problems the procedure stopped after at most 5 iterations.

We can use a similiar approach to derive an approximation for assembly systems with more than two lines. In the case where there are k lines in the system, assembly will not begin unless all k lines have at least one job at the assembly machine. To calculate the delay that jobs from line 1 experience waiting for jobs from other lines, we now assume that lines 2,...,k are independent closed queueing networks. To illustrate the nature of the calculations

involved, we consider an example with 3 lines. We can again condition on the position of the jobs in lines 2 and 3 closest to assembly at the time a job from line 1 arrives to the assembly machine. Denote the position of the job closest to assembly in line j as Z_j. For example, if in line 2 the job closest to the assembly machine is at machine 3, then $Z_2 = 3$. Similarly, if in line 3, there is already a job at assembly, then $Z_3 = m_3 + 1$. (Clearly, $Z_j = i$ corresponds to $\{N_{j,i} > 0, \sum_{p=i+1}^{m_2+1} N_{j,p} = 0\}$, where $N_{j,i}$ is the number of jobs in machine (j,i)). Then, we can write

$$E[W_1] = \sum_{i=1}^{m_2+1} \sum_{j=1}^{m_3+1} E[W_1|Z_2 = i, Z_3 = j]P(Z_2 = i, Z_3 = j) \tag{4.10}$$

Since, we are assuming that lines 2 and 3 along with the assembly machine behave as independent closed queueing networks, we have

$$E[W_1] \simeq \sum_{i=1}^{m_2+1} \sum_{j=1}^{m_3+1} E[W_1|Z_2 = i, Z_3 = j]P(Z_2 = i)P(Z_3 = j) \tag{4.11}$$

The probabilities in (4.11) can be computed as in the 2-line case. The calculation of the conditional expectations is more complicated. If $Z_2 = m_2 + 1$, then we are left with the case where jobs from line 1 are waiting for jobs from line 3 only, and then using the same approximation as in the two line case, we can write

$$E[W_1|Z_2 = m_2 + 1, Z_3 = j] \simeq \sum_{p=j}^{m_3} x_{3,p} \tag{4.12}$$

$$E[W_1^2|Z_2 = m_2 + 1, Z_3 = j] \simeq \sum_{p=j}^{m_3} \sigma_{3,p}^2 + (\sum_{p=j}^{m_3} x_{3,p})^2 \tag{4.13}$$

The case where $Z_3 = m_3 + 1$ is identical.

In the case where $Z_2 = i$ and $Z_3 = j$, and $i < m_2 + 1$, and $j < m_3 + 1$, getting an estimate of the first two moments of the delay that a job from line 1 experiences, waiting for the jobs from other lines before it can be assembled, is more complicated. Since the only information that we have on the processing times at each machine is the mean and the variance, this problem reduces to a problem of estimating the mean and variance of the maximum of k random variables each with known mean and variance. This problem is a difficult theoretical problem and even though some results have been developed for bounding

the mean (e.g., Birge and Dula, 1991), we are not aware of approximations for the variance. However, we note that in our assembly system, we assumed that each machine's processing time has a coefficient of variation less than 1. Note that the expression for the expected delay that a job from line 1 experiences given the positions of the other jobs in the other lines involves the convolution of processing times and convolution of random variables results in random variables with lower coefficients of variation. To obtain a rough approximation of the expected delay, we disregard the variability of the processing times and assume that they are deterministic. This results in the following expressions for the delay:

$$E[W_1|Z_2 = i, Z_3 = j] \simeq \max\{\sum_{p=i}^{m_2} x_{2,i}, \sum_{p=j}^{m_3} x_{3,i}\} \quad i \neq m_2 + 1, j \neq m_3 + 1 \qquad (4.14)$$

$$E[W_1^2|Z_2 = i, Z_3 = j] \simeq (E[W_1|Z_2 = i, Z_3 = j])^2 \quad i \neq m_2 + 1, j \neq m_3 + 1 \qquad (4.15)$$

Even though equations (4.10-4.15) provide a very rough approximation of the delay experienced by a job in line 1 waiting for jobs from other lines at assembly, our simulation results, which we report in the next section, indicate that it is adequate for the purposes of estimating the throughput of the system, and average WIP levels at each station. As in the 2-line case, just as jobs in line 1 experience a delay waiting for jobs from other lines, jobs in the other lines will experience delays as well. Hence, we again formulate a recursive procedure which we present below:

Procedure for computing θ_{ap} (k lines)

1. Let $h = argmin_r\theta\{F_r/F_A/n_r\}$. Renumber line h as line 1. Let $\theta_1 = min_r\theta\{F_r/F_A/n_r\}$. Renumber the other lines arbitrarily. Let $EW_i = 0$ and $Var[W_i] = 0$ for all $i = 1,\ldots,k$.

2. For $i = 2,\ldots,k$, compute the new values of EW_i and $Var[W_i]$ (denote the new values by EW_i' and $Var[W_i']$) from the closed queueing networks $\{F_r/F_A + W_r/n_r\}$ $r = 1,\ldots,k, r \neq i$, using equations (4.10)-(4.15).

3. Let $EW_i = EW_i'$ and $Var[W_i] = Var[W_i']$ for $i = 2,\ldots,k$

4. Compute EW_1' and $Var[W_1']$ using $\{F_r/F_A + W_r/n_r\}$ $r = 2,\ldots,k$ and equations (4.10)-(4.15).

5. Let $EW_1 = EW_1'$, and $Var[W_1] = Var[W_1']$

6. Let $\theta_{ap} = \theta\{F_1/F_A + W_1/n_1\}$. If $|\theta_{ap} - \theta_1| < \epsilon$ then stop. Else, $\theta_1 = \theta_{ap}$. Go to 2.

In this case, the algorithm makes use of the Shantikumar and Gocmen approximation k times in each iteration. However, as we describe in the next section, we found that the convergence of the approximation procedure was very rapid in all of the examples that we tested.

The above approximation procedures can also be used to obtain an estimate of the average WIP levels at each station. To do this, we note that we are approximating each line r by the closed queueing network $\{F_r/F_A + W_r/n_r\}$, generated in the last iteration of the above algorithm. Hence, applying the Shantikumar and Gocmen approximation procedure for closed queueing networks to $\{F_r/F_A + W_r/n_r\}$, we can obtain an estimate of the average WIP levels at each station in line r. We test the accuracy of this procedure in the next section.

5 Computational Results

In this section, we report the results of our simulation study in which we tested the performance of our approximation over a range of cases. To test our approximation, we generated a variety of problems with 2,3 and 4 lines, and compared the throughput of the system from simulation, θ_s, with our approximation results. Table 1 summarizes the scenarios that we tested in our simulation study. They are representative of the range of scenarios that could be observed in practice. They include cases with balanced and unbalanced fabrication lines, fast or slow assembly operations, and coefficient of variation of processing times ranging from 0 to 1. For each case, we examined the accuracy of our approximation for a

variety of WIP allocations. To simulate the assembly systems, we made use of a MOR-DS (Curry et al., 1989) program. For each WIP allocation, we simulated the system 10 times for 52000 time units, and the first 2000 time units of each run was truncated. We recorded the throughput each time and the average of 10 values gave us θ_s. Each simulation run (10 values) lasted about 15 minutes on a 486 machine, while the computation involved in our approximation took negligible time (less than 1 second for each example we considered). In fact, convergence was so quick that the value of the approximation after 2 iterations was always very close to the value of the approximation when $\epsilon < 0.001$.

The first three examples that we considered were balanced systems. In the first case, there were 2 fabrication lines and each line had 3 machines. All the fabrication machines and the assembly machine had processing time distributions that were Erlang-2 with mean 1. The simulation and approximation results are displayed in Table 2. The approximation behaved very well in this case and the accuracy was within 1.2 %. We next tested the approximation on balanced systems that were more and less variable than the one in Example 1. The system in Example 2 was identical to the system in Example 1. However, all the processing times had Erlang-4 distributions with mean 1. Hence, the variability was lower in this case. The approximation overestimated the throughput for all the WIP levels in this case. However, the results were still very good and the maximum error was 3.2 %. The system in Example 3 was again the same as in Example 1. However, in this case the first machine in each line was deterministic, while all the other machines in the fabrication lines and the assembly machine had exponential processing time distributions. The mean processing time was 1 for all machines. The approximation behaved very well again and the maximum error was about 1.4 %.

The next three examples tested the effect of the location of the bottleneck on the approximation. In Example 4, we again considered a system with 2 lines and 3 machines in each line. In this case, the first machine in each line had an Erlang-2 distribution with mean 1, the second machine in each line had an Erlang-4 distribution with mean 1, and

Example	Number of Fabrication Lines	Location of Bottleneck	Number of Machines in Each Line
1-3	2	Balanced	3-3
4	2	Assembly	3-3
5-6	2	Fabrication	3-3
7	2	Fabrication	4-3
8	3	Fabrication	4-4-4
9	3	Fabrication	2-3-4
10	4	Balanced	3-3-3-3

Table 1: Description of Examples

n_1	n_2	θ_s	θ_{ap}	%err
3	3	0.511	0.516	1.0
4	4	0.604	0.611	1.2
5	5	0.673	0.677	0.6
6	6	0.722	0.725	0.4
4	6	0.646	0.649	0.5
5	7	0.702	0.707	0.7
8	8	0.782	0.789	0.9
10	10	0.827	0.828	0.1

Table 2: Results for Example 1

n_1	n_2	θ_s	θ_{ap}	%err
3	3	0.575	0.590	2.6
4	4	0.688	0.706	2.6
5	5	0.759	0.775	2.1
6	6	0.806	0.819	1.6
4	6	0.724	0.747	3.2
5	7	0.788	0.803	1.9
8	8	0.864	0.870	0.7
10	10	0.895	0.900	0.6

Table 3: Results for Example 2

n_1	n_2	θ_s	θ_{ap}	%err
3	3	0.476	0.475	-0.3
4	4	0.555	0.557	0.4
5	5	0.614	0.618	0.7
7	7	0.693	0.703	1.4
3	5	0.517	0.514	-0.6
4	6	0.591	0.590	-0.2
9	9	0.756	0.757	0.1
6	6	0.660	0.666	0.9

Table 4: Results for Example 3

the processing at the third machine in each line was deterministic and lasted 1 time unit. The assembly machine had an exponential distribution with mean 2.5. The approximation again behaved very well, and the maximum error was only 1.7 %. Next, we exchanged the assembly machine with the third machine in line 1 of Example 4 to create an example where the bottleneck is in fabrication. That is, Example 5 was identical to Example 4 except that the third machine in line 1 is exponential with mean 2.5, and the assembly machine is deterministic with duration 1. The results for this example are displayed in Table 6. The maximum error was greater in this case. However, the approximation was still very close to the simulation value for nearly all WIP allocations. Notice also that a comparison of Table 5 and Table 6 shows how the placement of the bottleneck in fabrication improved the throughput for all WIP levels. We return to this issue at the end of this section.

The next example that we considered (Example 6) had an assembly machine that was considerably faster than all the other machines in the network. The first machine in both fabrication lines had an Erlang-2 distribution with mean 1, and the second and third machines in both lines were exponential with mean 1. The assembly machine was exponential with mean 0.3. The results for Example 6 are displayed in Table 7. Again, even though the maximum error was 3.8 %, the approximation behaved very well for nearly all WIP allocations.

In Examples 1-6, there were two fabrication lines and the two fabrication lines were

n_1	n_2	θ_s	θ_{ap}	%err
2	2	0.279	0.275	-1.5
4	2	0.295	0.291	-1.4
3	3	0.343	0.339	-1.2
4	4	0.373	0.372	-0.3
6	6	0.394	0.395	0.3
6	4	0.381	0.378	-0.8
4	3	0.352	0.347	-1.4
3	2	0.292	0.287	-1.7

Table 5: Results for Example 4

n_1	n_2	θ_s	θ_{ap}	%err
2	2	0.288	0.278	-3.5
4	2	0.363	0.364	0.3
3	3	0.346	0.346	0.0
4	4	0.381	0.378	-0.8
6	6	0.398	0.397	-0.3
6	4	0.397	0.396	-0.3
4	3	0.379	0.375	-1.0
3	2	0.338	0.334	-1.2

Table 6: Results for Example 5

n_1	n_2	θ_s	θ_{ap}	%err
3	3	0.521	0.501	-3.8
4	4	0.602	0.586	-2.7
6	6	0.704	0.699	-0.8
8	8	0.764	0.768	0.5
10	10	0.807	0.813	0.7
5	7	0.689	0.684	-0.8
7	9	0.766	0.760	-0.8

Table 7: Results for Example 6

identical. To see whether having fabrication lines that are not identical or having more than 2 fabrication lines affects the accuracy of the approximation, we generated Examples 7-10. In Example 7, the first fabrication line had 4 machines and the second had 3 machines. The distributions for the machines in the first line were respectively Erlang-2 with mean 1, exponential with mean 1.5, deterministic with duration 0.5 and exponential with mean 0.2. The second line had 3 machines with distributions Erlang-2 with mean 1, Erlang-2 with mean 1.5 and Erlang-2 with mean 0.5. Finally, the assembly station also had an Erlang-2 distribution with mean 1. The results for Example 7 are displayed in Table 8. Again, the results were within 3 % of simulation values.

Examples 8 and 9 have three fabrication lines, while Example 10 has 4 fabrication lines. Example 8 is system with 3 lines and 4 machines in each line. All machines except the assembly machine have Erlang-2 distributions with mean 1, while the assembly machine has an Erlang-2 distribution with mean 2. The results were again pretty good and the maximum error was 2.0 %. In Example 9, the first line had 2 machines and the second and third lines had, respectively, 3 and 4 machines. The processing time distributions (and means) for line 1 were: Deterministic (0.75), Exponential (1.5), while the distributions (and means) for line 2 were: Deterministic (1), Erlang-2 (1), Exponential (1). Finally, for line 3, they were: Deterministic (1), Erlang-4 (1), Erlang-2 (1), and Exponential (1). The assembly machine was exponential with mean 0.8. The approximation had a slightly harder time in this case, though the results were still within 3.5 %. In Example 10, there were 4 fabrication lines with 3 machines in each fabrication line. All the machines in the fabrication line as well as the assembly machine had Erlang-2 processing times with mean 1. Despite the increase in the number of lines, the approximation did very well, and the maximum error was only 2.1 %.

These ten examples are representative of our experience that the approximation for throughput behaves very well for a wide range of systems. In all cases, the maximum error was within 3.5 %. Also, factors such as the location of the bottleneck, the number

n_1	n_2	θ_s	θ_{ap}	%err
2	2	0.355	0.344	-3.1
3	3	0.454	0.451	-0.7
4	4	0.519	0.520	1.9
6	6	0.588	0.594	1.0
5	4	0.540	0.546	1.1
4	3	0.488	0.485	-0.7
3	2	0.395	0.385	-2.5

Table 8: Results for Example 7

n_1	n_2	n_3	θ_s	θ_{ap}	%err
3	3	3	0.342	0.349	2.0
4	4	4	0.411	0.414	0.7
5	5	5	0.455	0.454	-0.3
6	6	6	0.476	0.477	0.2
3	4	5	0.377	0.378	0.3
5	6	7	0.467	0.466	0.3

Table 9: Results for Example 8

n_1	n_2	n_3	θ_s	θ_{ap}	%err
3	3	3	0.450	0.435	-3.4
5	5	5	0.579	0.590	1.9
7	7	7	0.639	0.644	0.8
5	4	4	0.545	0.554	1.7
5	6	5	0.588	0.595	1.2
4	5	4	0.527	0.538	2.1
4	4	6	0.547	0.533	-2.6

Table 10: Results for Example 9

n_1	n_2	n_3	n_4	θ_s	θ_{ap}	%err
2	2	2	2	0.347	0.350	0.9
3	3	3	3	0.469	0.479	2.1
4	4	4	4	0.564	0.574	1.8
5	5	5	5	0.632	0.640	1.3
9	9	9	9	0.781	0.784	0.4

Table 11: Results for Example 10

of lines, and whether the lines are balanced or not did not have a significant effect on the quality of the approximation. The quality of the approximation did not get worse as the number of lines were increased, although the amount of computation involved increased. Yet, even when examples with 7 or 8 lines were considered, convergence was very quick, and we obtained answers in just a few seconds on a 486 computer.

As we pointed out in the previous section, our approximation procedure can also be used to estimate the average number of jobs at each station. For each example presented above, we also tested how our approximation performed when estimating average WIP levels at each station. As a typical example, we present in Table 12 the average WIP levels at each station in line 1 of the network of Example 7. We let n_{1j}^{s} be the average WIP level in line 1, machine j obtained by simulation, and we let n_{1j}^{ap} be the average WIP levels obtained by our approximation. Similarly, n_{1A}^{s} and n_{1A}^{ap} denote the average WIP levels (obtained by simulation and approximation) from line 1 at the assembly machine. As it can be seen from Table 12, the approximation was successful in estimating the average WIP levels at each machine. The results for Line 2 were similiar and this case is representative of our experience with the accuracy of the approximation in estimating average WIP levels at each machine.

We also tested the approximation in Duenyas and Hopp (1992a) for the throughput of assembly systems, and compared it against the approximation developed in this paper. The Duenyas and Hopp approximation (we will refer to it as (DHAPP)) was developed under the assumption that all processing times are exponential. Hence, it requires only mean processing time values as input. We have found that the two approximations give comparable results when all processing times are indeed exponential. However, when processing times are not exponential, (DHAPP) does not perform well. In particular, for systems where the processing times are less variable than the exponential distribution, (DHAPP) tends to underestimate throughput significantly. As an example, consider a 2-line assembly system where both lines have 3 machines with mean processing times of 2, and the assembly ma-

n_1	n_2	n_{11}^s	n_{11}^{ap}	n_{12}^s	n_{12}^{ap}	n_{13}^s	n_{13}^{ap}	n_{14}^s	n_{14}^{ap}	n_{1A}^s	n_{1A}^{ap}
2	2	0.40	0.39	0.67	0.65	0.19	0.18	0.07	0.07	0.67	0.71
3	3	0.58	0.58	1.07	1.09	0.25	0.24	0.09	0.10	1.01	0.99
4	4	0.71	0.75	1.60	1.59	0.29	0.29	0.10	0.11	1.30	1.26
6	6	0.96	1.00	2.82	2.77	0.34	0.35	0.12	0.13	1.77	1.74
5	4	0.75	0.84	2.04	2.03	0.31	0.32	0.11	0.12	1.79	1.69
4	3	0.63	0.70	1.45	1.47	0.28	0.28	0.10	0.11	1.54	1.45

Table 12: Approximation of WIP Levels

n_1	n_2	θ_s	θ_{ap}	% dif	DHAPP	% dif
2	2	0.187	0.190	1.6	0.153	-18.2
3	3	0.254	0.264	3.9	0.197	-22.4
5	5	0.315	0.322	2.2	0.252	-20.0
2	4	0.205	0.211	2.9	0.169	-17.6
3	5	0.270	0.280	3.7	0.240	-11.1
4	6	0.307	0.314	2.3	0.287	-6.5

Table 13: Comparison of DHAPP and our approximation

chine has a mean processing time of 3. However, the processing times in the first, second and third machines in each line have, respectively, Erlang-2, Erlang-4, and deterministic distributions. The assembly machine has an Erlang-4 distribution. The performance of the approximation derived in this paper, θ_{ap}, and of (DHAPP) for this example is displayed in Table 13. Not surprisingly, DHAPP underestimates the throughput in every case, and for each case considered the performance of DHAPP is very bad. These results clearly indicate that unless the processing times are exponential, the approximation developed in this paper outperforms DHAPP.

We now return to the issue of the effect of the location of the bottleneck on the throughput of the assembly system. The only difference between Examples 4 and 5 was that we interchanged the assembly machine with one of the fabrication machines, thereby moving the bottleneck from assembly to fabrication. In Table 14, we tabulate the percentage increase in throughput that this exchange resulted in. θ_1 denotes the throughput of the system when the bottleneck is at assembly, and θ_2 denotes the throughput of the system

when the bottleneck is at fabrication. The results indicate that this exchange could lead to an increase in throughput which can be as high as 23 %. We note that in all the examples that we tested, we observed this behavior. We note that for systems where all the processing times have the exponential distribution, Duenyas and Hopp (1992a) made a similiar observation that exchanging the assembly machine with a fabrication machine that has a lower mean processing time improves throughput. Our simulation results indicate that this result generalizes to systems with general processing times. This observation has implications for capacity allocation. For example, if the bottleneck is at assembly, and a manager has the option to move a worker from fabrication to assembly, thereby decreasing the capacity at fabrication, but increasing it at assembly, the above observation indicates that the exchange may improve throughput.

In our simulation experiments, we also observed that even if the bottleneck machine and one of the fabrication machines have the same mean processing time, if the fabrication machine is less variable than the bottleneck machine, exchanging them improves the throughput. As an example of this behavior, consider the assembly system in Example 3. In that example, the first machine in both lines was deterministic with duration 1, while the second and third machines in both lines as well as the assembly machine was exponential with mean 1. Now, assume that we exchange the assembly machine with the first machine in line 1. In Table 15, we tabulate the percentage improvement that this exchange causes for different WIP levels. Although the improvement in throughput was less pronounced in this case than in the previous example, this exchange still improved the throughput as much as 6.0 %.

6 Conclusions and Further Research

In this paper, we derived approximations for the throughput and average WIP levels for an assembly-like queueing system with general processing time distributions. We conducted a simulation study which indicates that our approximation is robust under a wide variety of

n_1	n_2	θ_1	θ_2	% dif
2	2	0.279	0.288	3.2
4	2	0.295	0.363	23.0
4	4	0.373	0.381	2.1
6	6	0.394	0.399	1.3
6	4	0.381	0.397	4.2
4	3	0.352	0.379	7.7

Table 14: Effect of Location of Bottleneck

n_1	n_2	θ_1	θ_2	% dif
3	3	0.476	0.484	1.6
4	4	0.555	0.563	1.4
5	5	0.614	0.628	2.3
7	7	0.693	0.710	2.5
5	3	0.517	0.548	6.0
6	4	0.591	0.623	5.4

Table 15: Effect of the Variance at Assembly

conditions. The approximation developed in this paper can be used, in conjunction with simulation, to aid decision makers in the configuration of fabrication/assembly lines. We also observed that a bottleneck at assembly limits throughput more than an equivalent bottleneck in fabrication. Exchanging (in terms of capacity, not functionality) a machine at assembly with a faster or less variable machine at fabrication increases throughput. Further research should address the following questions:

1. In this paper, we focused on the throughput of the assembly system. Further research should characterize the cycle time variance and the variance of the cumulative output process until a fixed time t. This problem was addressed for a tandem CONWIP system in Duenyas and Hopp (1990), however it has not been addressed for assembly systems.

2. In this paper, we developed approximations for throughput for an assembly system under the CONWIP protocol. Further research should characterize the throughput of assembly systems under different release mechanisms such as kanban. Furthermore,

further research is needed to compare the performance of different release mechanisms for assembly systems.

Bibliography

Adan, I., and J. Van der Wal., 1989. "Monotonicity of the throughput in single server production and assembly networks with respect to buffer sizes," in *Queueing Networks with Blocking*, eds. H.G. Perros and T. Altiok, North Holland, Amsterdam, pp. 345.

Ammar, M.H. 1980. "Modelling and analysis of unreliable manufacturing assembly networks with finite storage," MIT Laboratory for Information and Decision Sciences, Report LIDS-TH-1004.

Ammar, M.H., and S.B. Gershwin, 1989. "Equivalence relations in queueing models of fork/join queueing networks with blocking," *Performance Evaluation*, **19**, 233.

Bacelli, F, A.M. Makowski, and A. Shwartz, 1989. "The fork-join queue and related systems with synchronization constraints: Stochastic ordering, approximations, and computable bounds," *Advances in Applied Probability*, **21**, 629.

Baker, K.R., S.G. Powell, and D.F. Pyke, 1990. "Buffered and Unbuffered Assembly Systems with Variable Processing Times," *Journal of Manufacturing and Operations Management*, 3 200-223.

Baker, K.R., S.G. Powell, and D.F. Pyke, 1993. "Optimal Allocation of Work in Assembly Systems," *Management Science*, **39**, 101.

Bhat, U.N., 1986. "Finite capacity assembly-like queues," *Queueing Systems: Theory and Applications*, **1** 85.

Birge, J.R., and J.H. Dula. 1991. "Bounding Separable Recourse Functions with Limited Distribution Information," *Annals of Operations Research* **30** 277-298.

Bonomi, F., 1987. "An approximate analysis for a class of assembly-like queues," *Queueing Systems: Theory and Applications* **1**, 289.

Buzen, J.P. 1973. "Computational Algorithms for Closed Queueing Networks with Wxponential Servers," *Communications of the ACM* **16**, 527.

Curry, G.L., B.L. Deuermeyer, and R.M. Feldman, 1989. *Discrete Simulation: Fundamentals and Microcomputer Support*, Holden-Day, Oakland.

Dallery, Y., and S.B. Gershwin. 1992, "Manufacturing flow line systems: a review of models and analytical results," *Queueing Systems, Theory and Applications,* **12,** 3.

Duenyas, I., and W.J. Hopp, 1990. "Estimating Variance of Output from Cyclic Exponential Queueing Systems," *Queueing Systems: Theory and Applications* **7**, 337.

Duenyas, I., and W.J. Hopp, 1992a. "Estimating the Throughput of an Exponential CONWIP Assembly System," to appear in *Queueing Systems: Theory and Applications.*

Duenyas, I., and W.J. Hopp, 1992b. "CONWIP Assembly with Deterministic Processing and Random Outages," *IIE Transactions* **24** No.4, 97.

Ehteshami, B., R.G. Petrakian, and P.M. Shabe, 1992. "Trade-Offs in Cycle Time Management: Hot Lots," *IEEE Transactions on Semiconductor Manufacturing* **5**, 94.

Gershwin, S.B., 1991. "Assembly/Disassembly Systems: An Efficient Decomposition Algorithm for Tree-Structured Networks," *IIE Transactions,* **23**, 302.

Harrison, J.M., 1973. "Assembly-like queues," *Journal of Applied Probability* **10**, 354.

Hopp, W.J., and J.T. Simon. 1989. "Bounds and Heuristics for Assembly-Like Queues," *Queueing Systems: Theory and Applications* **4** 137.

Lipper, E.H., and B. Sengupta. 1986. "Assembly-like queues with finite capacity: bounds, asymptotics and approximations." *Queueing Systems: Theory and Applications* **1** 67.

Mascolo, M.D., R. David, and Y. Dallery,1991. "Modelling and Analysis of Assembly Systems with Unreliable Machines and Finite Buffers," *IIE Transactions,* **23**, 315.

Monden, Y. 1983. *Toyota Production System* Industrial Engineering and Management Press.

Shantikumar, J., and M. Gocmen, 1983. "Heuristic Analysis of closed queueing networks," *International Journal of Production Research* **21** 675.

Spearman, M.L., W.J. Hopp, and D.L. Woodruff. 1989. "A Hierarchical Control Architecture for Constant Work-In-Process (CONWIP) Production Systems," *Journal of Manufacturing and Operations Management* **2**, 147.

Spearman, M.L., D.L. Woodruff, and W.J. Hopp. 1990. "CONWIP: A Pull Alternative to Kanban." *International Journal of Production Research* **28**, 879.

Spearman, M.L., and M.A. Zazanis. 1992. "Push and Pull Production Systems: Issues and Comparisons." *Operations Research* **40**, 521-532.

Whitt, W. 1983. "The Queueing Network Analyzer," and "The Performance of the Queueing Network Analyzer." *Bell Systems Technical Journal,* **62**, 2799-2843.

Whitt, W. 1984. "Open and Closed Models for Networks of Queues," *Bell Laboratories Technical Journal,* **63** 1911.

UNIVERSITY OF MICHIGAN
3 9015 04735 3290