Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

DETERMINING JOB COMPLETION TIME DISTRIBUTIONS IN STOCHASTIC PRODUCTION ENVIRONMENTS

C. Nirmal Hasan

Dept. of Ind. Eng. & Mgmt. Sciences
Northwestern University
Evanston, Illinois 60208, U.S.A.

Mark L. Spearman

School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332, U.S.A.

ABSTRACT

As responsiveness becomes more of a competitive issue, knowledge of completion times of jobs and orders on the shop floor is key to the success and even survival of many manufacturing firms. In this paper, we present two approaches — a fast sample path generation technique and a numerical technique — to determine the distribution of completion times of jobs in stochastic production environments. We compare our results with those obtained from conventional discrete-event simulation and also compare speed of execution. We also show that the distribution obtained from the numerical integration technique provides a lower bound to the actual distribution of the completion times.

1 INTRODUCTION

In this age of global competition, the ability to respond quickly to the needs of customers is key to the very survival of many manufacturing firms. Frequently, a manufacturing firm has to negotiate the delivery dates with customers before accepting orders. Under such circumstances it would be invaluable for the firm to have an accurate estimate of when it could possibly deliver the products, so as to set reasonable delivery dates. In other words, the firm requires ability to predict the times of completion of various jobs and orders, including those that are yet to be released to the shop-floor.

At other times, a customer would like to know the status of his or her order(s). Again, a knowledge of the completion times of jobs on the shop-floor would be invaluable. Furthermore, if it was possible for the firm to know sufficiently in advance that some orders would be delayed, then the firm could inform the concerned customers and possibly re-negotiate the delivery dates, thereby leading to improved customer relations.

Knowledge of completion times of the jobs could also be used in determining the optimal release times for future orders such that the firm operates with short cycle-times and low work-in-process (WIP). This in turn would lead to improved quality of the products, by reducing the time between defect creation and defect detection.

In a typical manufacturing environment, the processing times at different process centers are not deterministic. Likewise, process centers may be subject to random outages. Further, several jobs compete for the same resources leading to queueing of jobs and congestion at process centers. Finally, the system is in a constant state of flux, with new orders being released and existing order requirements changing. All these combine to make the determination of the completion times of jobs extremely difficult.

The contributions of this research is a method that, given the current WIP situation on the shop floor and the release times of jobs, determines with reasonable speed and accuracy, the distribution of completion times of jobs at different process centers. Such a system enables the firm to

- Predict completion times of various jobs and customer orders
- Determine the feasibility of different schedules
- Set and modify due-dates for jobs for any desired service level
- Quote lead times for customers

Researchers investigating job shops have traditionally assumed that determination of due-dates for jobs was exogenous to the shop. Consequently, most of the research has been focused on developing procedures that optimize various due-date performance measures.

Eilon and Chowdhury (1976) were the first to propose that due-dates for jobs may be determined by

the job shop internally and suggested a due-date setting method that considers the average waiting time and the number of jobs that would be encountered by jobs entering the shop floor. Other due-date setting methods that consider the current workload have also been investigated (Bertrand 1983, Fry, Philipoom and Markland 1989, Weeks 1979, Baker 1984), as have cost based models (Weeks and Fryer 1977, Seidmann and Smith 1981).

Service level was first used by Taylor and Moore (1982) in the context of setting due-dates. They proposed that a knowledge of flow times of jobs should be used for negotiating due-dates with customers. Bookbinder and Noor (1985) also linked the setting of due-dates with service-level constraints and shop floor congestion information.

Recently, Wein (1991) unified the conflicting goals of minimizing due-date lead times and maximizing service level, by stating the problem as one of minimizing the weighted sum of cycle times subject to a minimum service level constraint. Wein and Chevalier (1992) propose a two step approach to scheduling the job shop—first use a job release policy that balances workload at bottlenecks and then determine due-dates using parameterized version of the rule proposed by Wein (1991). Spearman and Zhang (1994) use a formulation similar to Wein to determine an optimal lead time quoting policy.

The estimation of actual job flow times appears to have come up first in the context of finding an optimal due-date setting rule. Miyazaki (1981) modeled the job shop as a Jackson network (Jackson 1963) and obtained an exact formulation for the mean and an approximation for the variance of job flow times. Cheng (1985), and more recently Koulamas (1992), also using the Jackson network model, derived exact expressions for the first two moments of the job flow times. However, the models are not robust enough to approximate non-exponential systems. Furthermore, these results apply to steady-state conditions only.

While there exist some analytical solutions (Ross 1983, Kleinrock 1975) and approximations (Whitt 1983, Buzzacott and Shanthikumar 1985, Shanthikumar and Sumita 1988, Fleming and Simon 1991) for queueing systems in steady-state, transient analysis of queues have yielded relatively fewer results (see for e.g., Mori 1976, Kotiah 1978, Pack 1978).

Recent work by Saboo, Wang and Wilhelm (1989) uses a model description and solution approach similar to those described in this paper, but the goals of their research are substantially different from ours. Their work is discussed in more detail in the following section.

Karmarkar (1993) points out that determination of flow times for specific jobs and the use of such information in setting due-dates and determining release times has not been addressed adequately by researchers. Our research attempts to fill this void.

2 MODEL DESCRIPTION AND NOTATION

The system has a single part type (i.e., a single routing), and is an open serial, single-server queueing network. Other assumptions are

- Each job has a release time associated with it. At this point, it is assumed that the release time was determined externally, according to some production planning framework such as MRP.
- A process center may not be visited more than once in the routing, i.e., no re-entrant flows are allowed.
- Processing times for each *part* at a given process center in the routing follow distributions that are independent and identically distributed. Further, processing times of parts at one process center are independent of processing times of the parts at other process centers.
- At each process center, jobs are processed in strict FIFO sequence.
- Jobs cannot be pre-empted by other jobs while they are being processed by a machine at any process center.
- Resources do not fail, i.e., they have 100% availability
- There is no scrap loss in the system.
- The system operates 24 hours, 7 days a week, i.e., shift operation is not modeled.
- The system is empty at time $t = 0$, i.e., there is no WIP in the system.

We use the following notation:

I : Total number of process centers visited by each job in the routing. Since we assume that a job does not visit the same process center twice and that there is only one part type, this is also the total number of process centers in the system.

J : Total number of jobs that are present on the shop floor. Since we assume that the system is empty at time $t = 0$, this is also the number of jobs that are being released to the system.

r_j : Release time of job j, $j \in \{1, \ldots, J\}$. This is the time that job j becomes available for processing at the first process center in the routing.

S_{ij} : Random variable representing the processing time of job j at process center i, $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$.

C_{ij} : Random variable representing the completion time of job j at process center i, $i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$. The release time of a job is considered to be the completion time of that job at process center 0, i.e.,

$$C_{0j} = r_j \qquad \forall\, j$$

The time that the resource at process center i is ready to process the first job is considered to be the completion time of job 0 at that station and is taken to be 0, i.e.,

$$C_{i0} = 0 \qquad \forall\, i$$

3 COMPLETION TIME OF A JOB

It is obvious that for a particular process center i to start processing a particular job j, two conditions must be met—job j must be present at process center i, and the machine at i must be available to process job j. Since the system is a push type system, the former occurs when the job j completes processing at center $i-1$, while the latter occurs when center i completes processing the previous job $j-1$. Thus, the completion time of an arbitrary job j at an arbitrary process center i along the routing can be obtained as

$$C_{ij} = (C_{i-1j} \vee C_{ij-1}) + S_{ij} \qquad (1)$$

where $x \vee y = \max\{x, y\}$.

Recall that we assumed that each process center in the routing has exactly one machine. This assumption, along with the assumption that jobs may not pre-empt one another, gives this model an important property, namely, for any arbitrary job j, there is exactly one predecessor $j-1$, and this precedence relationship is maintained at every process center in the routing. Thus, for any job, we can uniquely identify the preceding job at every process center.

The structure of equation (1) suggests an iterative approach to obtaining the completion times of the jobs in the routing. Considering the jobs in the sequence in which they are released, we could use equation (1) iteratively to obtain the distribution of the completion time of each job in each process center, starting with the first process center in the job's routing. The basic mechanism of the algorithm is described below:

Algorithm A_0:

1. `Sequence the jobs in increasing order of their release times, with` $j = 1$ `for the job that is being released the earliest.`
2. `For` $i = 1, \ldots, I$
   `For` $j = 1, \ldots, J$
   `Obtain distribution of` C_{ij} `using equation (1)`
3. `Return the distribution of` C_{Ij} `as that of the completion time of job` j `in the routing.`

Equation (1) is reminiscent of the recursive equation used for the steady state waiting time distribution of a customer in a $G/G/1$ queuing system (Ross 1983, Kleinrock 1975), first established by Lindley in 1952. Recursive approaches similar to the proposed algorithm A_0 have been used for estimating the steady state waiting time distributions (Ross 1983, Sumita 1981). Pack (1978) suggested the use of Lindley's equation for obtaining the transient completion time distribution analytically, but was able to do so only in the case of $M/D/1$ and $D/M/1$.

It is important to note that while the approaches mentioned above have their similarities with our approach, they do not address the same issues as we do, namely that of determining the *transient* state distributions in a $G/G/1$ *open queue network*.

We propose two distinct approaches to implementing algorithm A_0:

1. Obtaining the distribution of C_{ij} empirically, by a fast sample-path generation technique.
2. Obtaining the distribution of C_{ij} directly, using numerical integration.

The following sections describe these methods in detail.

Saboo, Wang and Wilhelm (1989) use a model description that is identical to the one given by equation (1) and they too use a recursive approach. Beyond this point, however, their work differs from ours. Their performance measures are the expected makespan, delay in queues, station utilization and lot tardiness. Further, they use a bivariate normal approximation for the maximum of two random variables and consequently, require that the processing times at the process centers be be normally distributed. They state that approximation errors grow with the number of jobs and/or an increase in the

number of process centers. They also state that their algorithm substantially underestimates the variance of the finish times in most cases and conjecture that it is due to the normal approximation for the maximum of the two random variables.

3.1 Sample Path Generation Technique

It is certainly possible to obtain the cdf of the completion times of jobs using discrete-event simulation. In fact, Taylor and Moore (1982) propose such an approach for a relatively small job shop.

Since we are interested in the completion time distribution of the jobs, the stopping criterion would be the completion of the last job at the last process center. The release times of the jobs are known, and thus, the sequence in which the jobs arrive at the first process center is known. Further, with each process center having exactly one machine and with the assumption that the jobs do not preempt each other, the sequence of jobs at the first process center is exactly the sequence of jobs at every other process center.

The fast sample-path generation technique (SPGT) may be described as follows:

1. All the jobs are held in a job list in increasing order of their release times into the system.
2. At each process center, for each job, a realization of the processing time at that process center is generated and equation (1) is used to determine the completion time of that job.
3. This is continued until the completion time of the last job at the last process center has been determined.

Algorithm A_0 can be implemented using SPGT as follows:

Algorithm SE_1:

1. $c_{0jl} \leftarrow r_j, \ \forall j, l$
2. $c_{i0l} \leftarrow 0, \ \forall i, l$
3. For $i = 1, \ldots, I$
   For $j = 1, \ldots, J$
   For $l = 1, \ldots, L$
   Generate one realization of S_{ij}, say s_{ijl}
   Compute c_{ijl} using equation (1) as $c_{ijl} = (c_{i-1jl} \vee c_{ij-1l}) + s_{ijl}$
4. Tabulate the cdf of C_{Ij} for various values of time t

Note that while this technique may be similar to Monte Carlo techniques used for obtaining steady state waiting time distributions for $M/M/1$ queueing systems using Lindley's equation (Schmeiser and Song 1989, Rubinstein 1986), it is not identical. Further, the objectives are significantly different—transient measures for our approach versus steady state measures for the Monte Carlo approach.

Since the SPGT does not have events and the overhead associated with maintaining an event list, it is faster than conventional event driven simulation, while retaining the same accuracy.

One single run of either conventional simulation or SPGT results in a single realization of completion times for the jobs. Since we are interested in the transient behavior of the system, we need to replicate the run several times. This results is a series of completion times for each job from which the distribution (cdf) of the completion job can be determined, empirically.

Suppose we run L replicates and for each job j, we store c_{Ijl}, the completion time of the job at the last process center from replicate l. Since, we do not possess any knowledge of the theoretical distribution of C_{ij}, a natural estimate of the probability that C_{Ij} is less than or equal to t is given by

$$p_L^{Ij}(t) = \frac{1}{L} \sum_{l=1}^{L} 1_{(c_{Ijl} \le t)}$$

where $1_{(c_{Ijl} \le t)}$ is the indicator function that is 1 if $c_{Ijl} \le t$ and 0 otherwise. We need the following definition (Bickel and Doksum 1977, page 460):

Definition 1 A sequence of random variables $\{Z_L\}$ *converges* to the random variable Z *in probability* if $P\{|Z_L - Z| \ge \epsilon\} \to 0$ as $L \to \infty$ for every $\epsilon > 0$ and is denoted by $Z_L \xrightarrow{P} Z$.

To determine L, the number of replicates required to estimate the probabilities accurately, we proceed as follows. Note that $p_L^{Ij}(t)$ is an unbiased estimator of $p^{Ij}(t)$ with $Var(p_L^{Ij}(t)) = p^{Ij}(t)(1 - p^{Ij}(t))/L$, where $p^{Ij}(t) = P\{C_{Ij} \le t\}$. Further,

$$p_L^{Ij}(t) \xrightarrow{P} p^{Ij}(t)$$

From the Glivenko-Cantelli theorem (Fisz 1963, page 391), $\sup_x |p_L^{Ij}(t) - p^{Ij}(t)| \xrightarrow{P} 0$, i.e., the empirical distribution looks like the theoretical distribution for large values of L. With this, we may use the Kolmogorov goodness of fit test as

$$H_0: \quad p_L^{Ij} = p^{Ij}$$
$$H_1: \quad p_L^{Ij} \ne p^{Ij}$$

with the test statistic defined as

$$D_L = \sup_{-\infty<t<\infty}|p_L^{Ij}(t) - p^{Ij}(t)|.$$

We reject the null hypothesis for large values of D_L (Bickel and Doksum 1977). For a $(1-\alpha)\%$ confidence level, the critical value k_α is such that $P\{D_L \geq k_\alpha\} = \alpha$. For large $L > 80$ and $\alpha = 0.01$, $k_{0.01}$ is given by (Bickel and Doksum 1977, page 483)

$$k_{0.01} = \frac{1.628}{\sqrt{L} + 0.12 + \frac{0.11}{\sqrt{L}}}$$

If we require that the maximum absolute error in our estimate be 0.01, we set

$$\frac{1.628}{\sqrt{L} + 0.12 + \frac{0.11}{\sqrt{L}}} = 0.01$$

which yields $L = 26464.56$. For our purposes, we use $L = 26500$.

The SPGT was used on two test problems and the results compared with those from conventional discrete-event simulation. The simulation program used was an object-oriented (C++) version of MAC-Sim, a simulation program developed by one of the authors (Hasan 1991). All processing times were modeled as truncated normal distributions. The first test problem was a system with 2 process centers and with 4 jobs being released to the system while the second test problem was a system with 6 process centers and with 8 jobs being released to the system. A detailed description of the two test problems may be found in Section A. For each job, the completion time distribution at the last process center in the routing obtained from simulation and SPGT were plotted against each other, revealing no visible differences.

3.2 Numerical Technique

An approach different from simulation, is to compute the distribution functions of interest directly. Let F_{ij} denote the distribution of the completion time and G_{ij} denote the distribution of the processing time of job j at process center i. If C_{i-1j} and C_{ij-1} are independent, then, from equation (1), we may obtain the distribution of C_{ij} as

$$F_{ij}(t) = \int_0^t F_{i-1j}(t-x)F_{ij-1}(t-x)\,g_{ij}(x)\,dx \quad (2)$$

However, C_{i-1j} and C_{ij-1} are *not* independent and thus equation (2) is an approximation. Expanding C_{i-1j} and C_{ij-1} along the lines of equation (1) as

$$C_{i-1j} = (C_{i-2j} \vee C_{i-1j-1}) + S_{i-1j}$$

and

$$C_{ij-1} = (C_{i-1j-1} \vee C_{ij-2}) + S_{ij-1}$$

we see that C_{i-1j-1} appears in both the expansions, and it is the *same* realization of C_{i-1j-1}. We call this correlation due to C_{i-1j-1} as *correlation of the first order*. In fact, if we do a similar expansion for C_{i-2j}, C_{i-1j-1} and C_{ij-2}, we find that C_{i-2j} and C_{i-1j-1} are similarly correlated, as are C_{i-1j-1} and C_{ij-2}. It would be instructive, however, to see the effect of ignoring the correlation between C_{i-1j} and C_{ij-1} on the distribution of C_{ij}. As a first approximation, we assume independence between C_{i-1j} and C_{ij-1}, and use equation (2) to determine the completion time distributions.

Despite the assumption of independence it is not possible to solve equation (2) in closed form. The "max" operation on the distributions destroys any structure that we might use for the exact evaluation of the convolution integral in equation (2). Hence, we use numerical techniques. With the notation that

$$\Delta(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

The basic mechanism of this method as follows:

Algorithm N_1:

1. $F_{0j} \leftarrow \Delta(t - r_j),\ \forall\ j$
2. $F_{i0} \leftarrow \Delta(t),\ \forall\ i$
3. Represent the processing time pdf's $g_{ij1},\ \forall\ i$ numerically
4. For $i = 1, \ldots, I$
   For $j = 1, \ldots, J$
   For $l = 1, \ldots, L$
   Obtain cdf of max as the product of
   F_{i-1j} and F_{ij-1}
   Obtain cdf of C_{ij} by numerically evaluating equation (2)
5. Tabulate the cdf of C_{Ij} for various values of time t

The primary implementational issue is one of representing the various distributions in a form the lends itself to convolutions as well as the "max" operation. Distributions are represented as a *finite* sequence of N_X ordered pairs $(x_i, F_X(x_i))$ where $F_X(x_i)$ is the cdf of the random variable X, i.e., $F_X(x_i) = P\{X \leq x_i\}$. The spacing between the x_i's is the *lattice interval* denoted by τ. The value of τ is chosen to be the same for all the distributions used for a particular

model. To minimize truncation errors in representing distributions that have infinite supports, the points x_1 and x_{N_X} (and hence N_X) are chosen such that

$$x_1 = \sup\{n\tau : P\{X \leq n\tau\} \leq \alpha\}$$

and

$$x_{N_X} = \inf\{n\tau : P\{X \leq n\tau\} \geq 1 - \alpha\}$$

for $n = 0, 1, 2, \ldots$, and α sufficiently small.

Algorithm N_1 was tested against simulation and the resulting cdf's plotted were plotted together. For both these test problems, the numerical distribution of the completion time of the first job matched exactly with the empirical distributions obtained from simulation. This is only to be expected, since for the first job, the completion time is just the sum of all the processing times and there is no correlation effect.

However, the effect of correlation is apparent when we compare the distributions of the other jobs (Figures 1 and 2). In fact, the distribution computed numerically always seems to lie below the distribution obtained empirically, and the gap between the two distributions seems to increase with both the number of jobs and the number of process centers.

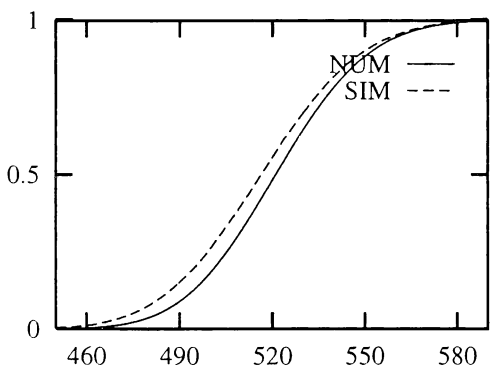

Figure 1: Distribution plot for job 4 (Test Problem 1)

The execution time on test problem 1 was 36.3 seconds (CPU time), whereas it was 1000.5 on the second test problem, which is more than conventional simulation (864.0 secs). The default value of the lattice interval length, τ, was initially taken as 0.5. When this was changed to 5, the resulting execution time on test problem 2 was 14.7 seconds, which is a dramatic improvement. The distributions computed using the two different values of τ were plotted against each other. The plots showed that there is no significant loss of accuracy despite a significant reduction in computation time (Figure 3). It appears that the interval length τ requires more investigation.

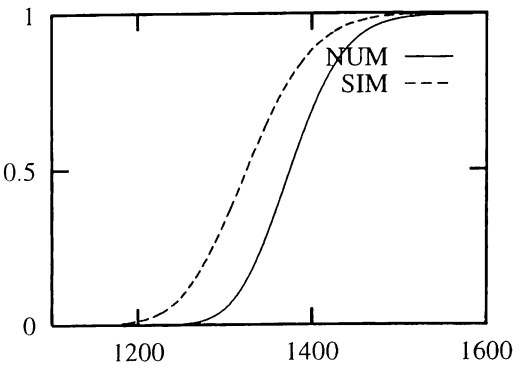

Figure 2: Distribution plot for job 8 (Test Problem 2)

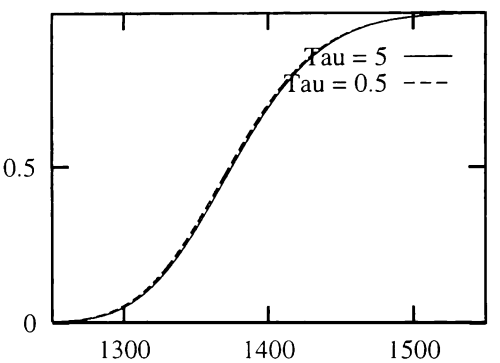

Figure 3: Comparison of distribution with $\tau = 0.5$ and $\tau = 5.0$ for job 8 (Test Problem 2)

Proposition 1 *Let $F_{ij}(t)$ be the distribution of C_{ij} and let $\hat{F}_{ij}(t)$ be the distribution of C_{ij} determined numerically, ignoring correlation between the various completion times. Then $\hat{F}_{ij}(t) \leq F_{ij}(t)$, $\forall\ i, j, t$.*

Consider a system with J jobs and I process centers. The very first job always sees an empty system as it proceeds along the routing. The completion time this job, C_{I1}, is simply the sum of the processing times at the various process centers. Thus $C_{I1} = \sum_{i=1}^{I} S_{i1}$ and $F_{I1}(t)$ is just the convolution of the process time distributions. Since there is no correlation effects for the first job, $F_{I1}(t) = \hat{F}_{I1}(t)$, $\forall\ t$. For job j at the first process center, $C_{1j} = (C_{0j} \vee C_{1j-1}) + S_{ij} = (r_j \vee C_{ij-1}) + S_{ij}$. Since release times of jobs are independent of the processing times of the jobs, $F_{1j}(t) = \hat{F}_{1j}(t)$, $\forall\ j, t$.

Now consider the completion time of the second job at the second process center,

$$C_{22} = (C_{12} \vee C_{21}) + S_{22}$$

Since the correlation effect is due to the "max" operation, let us ignore S_{22} and look at

$$\begin{aligned} D_{22} &= (C_{12} \vee C_{21}) \\ &= [(C_{02} \vee C_{11}) + S_{12}] \vee \\ &\quad [(C_{11} \vee C_{20}) + S_{21}] \end{aligned}$$

We need to use the *same* realization for C_{11} in both terms the variable appears in. However, since the numerical method does not actually use the expanded equation for C_{22}, it effectively use two *independent* realizations of C_{11} in the two terms in equation (3).

Let the actual distribution of D_{22} be H_{22} and the numerically estimated distribution of D_{22} be $\hat{H}_{22}$. Conditioning on the processing times, i.e., letting $S_{12} = a$ and $S_{21} = b$, we have the two *conditional* estimates as,

$$H_{22}^{ab}(t) = F_{02}(t-a)F_{11}(t-(a \vee b))F_{20}(t-b)$$

and

$$\hat{H}_{22}^{ab}(t) = F_{02}(t-a)F_{11}(t-a)F_{11}(t-b)F_{20}(t-b)$$

Note that F_{02}, which is the cdf of r_2, and F_{20}, which is the cdf of C_{02} are unit step functions, making the transition at r_2 and 0 respectively. We have retained them in the equation for the sake of generality.

If $a > b$,

$$H_{22}^{ab}(t) = F_{02}(t-a)F_{11}(t-a)F_{20}(t-b)$$

and if $b > a$, then

$$H_{22}^{ab} = F_{02}(t-a)F_{11}(t-b)F_{20}(t-b)$$

In either case, $\hat{H}_{22}^{ab} \leq H_{22}^{ab}$.

Since $\hat{H}_{22}^{ab}(t) \leq H_{22}^{ab}(t)$, $\forall\ a, b, t$, it follows that $\hat{H}_{22}(t) \leq H_{22}(t)$, $\forall\ t$. Thus, since $F_{22}(t) = H_{22}(t) * G_{22}(t)$ and $\hat{F}_{22}(t) = \hat{H}_{22}(t) * G_{22}(t)$, it follows that $\hat{F}_{22} \leq F_{22}$, $\forall\ t$.

Suppose now that we have the *exact* distributions $F_{i-1j}(t)$ and $F_{ij-1}(t)$ for some i and j. Then, from the algorithm N_1,

$$\hat{F}_{ij}(t) = (F_{i-1j}(t)F_{ij-1}(t)) * G_{ij}(t)$$

However, since $F_{i-1j}(t)$ and $F_{ij-1}(t)$ are correlated through the term $F_{i-1j-1}(t)$ (just as $F_{12}(t)$ and $F_{21}(t)$ are correlated through the term $F_{11}(t)$), $\hat{F}_{ij}(t)$ will be less than or equal to $F_{ij}(t)$. Note, however, that we obtain the distributions of C_{i-1j} and C_{ij-1} iteratively using algorithm N_1 and have only the *approximate* distributions $\hat{F}_{i-1j}(t)$ and $\hat{F}_{ij-1}(t)$. Thus,

$$\begin{aligned} \hat{F}_{ij}(t) &= (\hat{F}_{i-1j}(t)\hat{F}_{ij-1}(t)) * G_{ij}(t) \\ &\leq F_{ij}(t) \end{aligned}$$

□

Using conditioning arguments on the process times S_{i-1j} and S_{ij-1}, it is possible to remove the first order correlation effect and obtain a better approximation for $F_{ij}(t)$ as

$$\hat{F}_{ij}(t) = \hat{H}_{ij}(t) * G_{ij}(t)$$

where

$$\hat{H}_{ij}(t) = \int_0^t \int_0^t [F_{i-2j}(t-a)F_{i-1j-1}(t-(a \vee b)) \\ F_{ij-2}(t-b)]\, g_{i-1j}(a)\, g_{ij-1}(b)\, da\, db$$

However, we found that the speed of the algorithm degraded by several orders of magnitude for very little gain in accuracy (See Figure 4. "NUM1" refers to cdf obtained ignoring correlation while "NUM2" refers to cdf obtained after removing first order correlation. Test problem 3 was used).

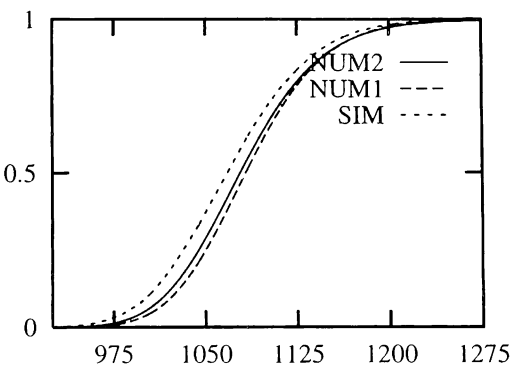

Figure 4: Comparison of distribution of job 8 (Test Problem 3)

In order to remove the correlation effects completely, we would need to expand equation (1) recursively until it could be written in terms of the C_{0j}'s and C_{i0}'s and the various S_{ij}'s. However, this appears to be completely intractable.

4 CONCLUSIONS

In this paper we have presented two techniques to determine the completion time distributions of jobs in stochastic production environments. The fast sample-path generation technique was show to be faster than conventional discrete-event simulation under the model assumptions specified. It was also

shown that the numerical technique gives progressively looser lower bounds on the actual distributions when correlations are ignored.

The assumptions for the model stated were indeed somewhat restrictive. Currently, research is being conducted on extending these techniques for use with models with fewer assumptions. We are also investigating the use of SPGT in determining optimal release times of jobs in stochastic production environments.

A APPENDIX: Description Of Test Problems

A.1 Test Problem 1

Number of jobs released: 4

	Release Time
*Job*1	0
*Job*2	0
*Job*3	0
*Job*4	10

Number of process centers: 2
Processing times:

	Distribution	Mean	Std. Dev
*station*1	Normal	100	10
*station*2	Normal	100	15

A.2 Test Problem 2

Number of jobs released: 8

	Release Times
*Job*1	0
*Job*2	0
*Job*3	0
*Job*4	10
*Job*5	10
*Job*6	10
*Job*7	20
*Job*8	20

Number of process centers: 6
Processing times:

	Distribution	Mean	Std. Dev
*station*1	Normal	100	10
*station*2	Normal	100	15
*station*3	Normal	100	30
*station*4	Normal	60	18
*station*5	Normal	75	20
*station*6	Normal	90	25

A.3 Test Problem 3

Number of jobs released: 8

	Release Times
*Job*1	0
*Job*2	0
*Job*3	0
*Job*4	10
*Job*5	10
*Job*6	10
*Job*7	20
*Job*8	20

Number of process centers: 3
Processing times:

	Distribution	Mean	Std. Dev
*station*1	Normal	100	10
*station*2	Normal	100	15
*station*3	Normal	100	30

REFERENCES

Baker, K. R., 1984. Sequencing Rules and Due-date Assignments in a Job Shop, *Management Science*, 30(9): 1093–2004.

Bertrand, J. W. M., 1983. Effect of Workload Dependent Due-dates on Job Shop Performance, *Management Science*, 29(7): 799–816.

Bickel, P. J., K. A. Doksum, 1977. *Mathematical Statistics: Basic Ideas and Selected Topics*, Holden-Day Inc. CA.

Bookbinder, J. H., A. I. Noor, 1985. Setting Job-shop Due-dates with Service-level Constraints, *Journal of Operational Research Society*, 36: 1017–1026.

Buzzacott, J. A., J. G. Shanthikumar, 1985. Approximate Queuing Models of Dynamic Job Shops, *Management Science*, 31(7): 870–887.

Cheng, T. C. E., 1985. Analysis of Flow-time in a Job Shop, *Journal of Operational Research Society*, 36: 225–239.

Eilon, S., I. G. Chowdhury, 1976. Due-dates in Job Shop Scheduling, *International Journal of Production Research*, Vol. 14.

Fisz, M., 1963. *Probability Theory and Mathematical Statistics*, 3rd edition, John Wiley & Sons, Inc., New York.

Fleming P. J., B. Simon, 1991. Interpolation Approximation of Sojourn Time Distributions, *Operations Research*, 39(2): 251–260.

Fry, T. D., P. R. Philipoom, R. E. Markland, 1989. Due-date Assignment in a Multi-stage Job Shop", *IIE Transactions*, 21(2).

Hasan, C. N., 1991. MACSim 1.0: User's and Programmer's Manual, Unpublished Master's thesis,

McCormick School of Engineering and Applied Sciences, Northwestern University, Evanston, Illinois.

Jackson, J. R., 1963. Job-shop Like Queueing Systems, *Management Science*, 10(1): 131–142.

Karmarkar, U. S., 1993. Manufacturing Lead Times, Order Release and Capacity Loading, In *Handbooks in OR & MS*, Vol. 4, 287–329, S. C. Graves *et al.*, eds., North-Holland, Amsterdam.

Kleinrock, L., 1975. *Queueing Systems, Volume I: Theory*, John Wiley & Sons, New York.

Kotiah, T. C. T., 1978. Approximate Transient Analysis of Some Queuing Systems, *Operations Research*, 26(2).

Koulamas C. P., 1992. Exact Expressions for the First Two Moments of Job Flow Time in a Job Shop, *Journal of Operational Research Society*, 43(9).

Miyazaki, S., 1981. Combined Scheduling System for Reducing Job Tardiness in a Job Shop, *International Journal of Production Research*, Vol. 19.

Mori, M., 1976. Transient Behavior of the Mean Waiting Time and Its Exact Forms in $M/M/1$ and $M/D/1$ Queues, *Journal of Operational Research Society, Japan*, 19(1).

Pack, C. D., 1978. The Output of a Multiserver Queueing System, *Operations Research*, 26(3): 492–509.

Ross, S. M., 1983. *Stochastic Processes*, John Wiley & Sons, New York.

Rubinstein, R. Y., 1986. *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*, John Wiley & Sons, New York.

Saboo, S., L. Wang, W. E. Wilhelm, 1989. Recursion Models For Describing and Managing the Transient Flow of Materials in Generalized Flowlines, *Management Science*, 35(6): 722–742.

Schmeiser, B., W. T. Song, 1989. Inverse-transformation Algorithms for Some Common Stochastic Processes, *Proceedings of the 1989 Winter Simulation Conference*, E. A. MacNair, K. J. Musselman, P. Heidelberger, eds.

Seidmann, A., M. L. Smith, 1981. Due-date Assignment for Production Systems, *Management Science*, 27(5): 571–580.

Shanthikumar, J. G., U. Sumita, 1988. Time Spent in a Dynamic Job Shop, *International Journal of Production Research*, 26(8): 1329–1352.

Spearman, M. L., R. Q. Zhang, 1994. Optimal Lead Times in Manufacturing Systems, working paper, Department of Industrial Engineering and Management Sciences, Northwestern University.

Sumita, U., 1981. Development of the Laguerre Transform Method for Numerical Exploration of Applied Probability Models, Unpublished Ph.D. Thesis, Graduate School of Management, University of Rochester, Rochester, New York.

Taylor III, B. W., L. J. Moore, 1982. Estimating Job Flow-times in a Job Shop for Contractually Negotiated Due-dates, *Journal of Operational Research Society*, Vol. 33.

Weeks, J. A., 1979. A Simulation Study of Predictable Due-dates, *Management Science*, 25(4): 363–373.

Weeks, J. K., J. S. Fryer, 1977. Methodology for Assigning Minimum cost Due-dates, *Management Science*, 23(8): 1362–1371.

Wein, L. M., 1991. Due-date Setting and Priority Sequencing in a Multi-class $M/G/1$ Queue, *Management Science*, 37(7): 1018–1033.

Wein, L. M., P. B. Chevalier, 1992. A Broader View of the Job-shop Scheduling Problem, *Management Science*, 38(7).

Whitt, W., 1983. The Queueing Network Analyzer, *Bell Systems Technical Journal*, 62(9): 2779–2815.

AUTHOR BIOGRAPHIES

C. NIRMAL HASAN is a graduate student in the Department of Industrial Engineering and Management Sciences at Northwestern University, working towards his Ph.D. degree. His research interests include simulation modeling and analysis of manufacturing systems and real-time mechanisms for production planning and control. He holds a Master's degree in Manufacturing Engineering from Northwestern University and a Bachelor's degree in Mechanical Engineering from Birla Institute of Technology and Science, India.

MARK L. SPEARMAN is Associate Professor of Industrial Engineering in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research is oriented toward developing production control strategies and techniques that are more effective than those offered by existing technologies (e.g., MRP, MRP II, OPT, kanban, etc.). He is co-author of the new book, *Factory Physics: The Foundations of Manufacturing Management* with Wallace J. Hopp.