Modeling Reentrant Manufacturing Systems with Inspection Stations

Y. Narahari and L.M. Khan, Indian Institute of Science, Bangalore, India

Abstract

Inspection stations are now an integral part of any manufacturing system and help track product quality and process performance. This paper considers reentrant manufacturing systems—such as semiconductor wafer fabrication systems—with inspections at various stages of processing. At the end of each inspection, three possibilities are assumed, namely accept, reject, or rework at some previous stage. Proposed are reentrant lines with probabilistic routing as models for such systems and presented is an efficient, approximate analytical technique based on mean value analysis to predict mean steady-state cycle times and throughput rates. The proposed method explicitly models scheduling policies used in reentrant lines and can be used for rapid performance analysis. Two applications of the analysis methodology are presented. First, it is shown how to compare alternative ways of locating inspection stations from a cycle time and throughput viewpoint. Next, by considering a realistic model of a wafer fab, evaluation is made of the effect of congestion at an inspection station on the cycle time and throughput of the wafer fab.

Keywords: Reentrant Lines, Inspection and Testing, Multiclass Queuing Models, Mean Value Analysis

Introduction

Global competitive pressures are forcing today's manufacturing companies to become more customer focused in terms of offering high-quality products and reduced product lead times. The recognition that product quality is a strategic asset has spurred factory managers to reexamine the role of on-line and off-line quality in product design and manufacturing. In spite of the best process control methods, it is impossible to eliminate defects altogether, hence inspection stations are essential.

Inspection stations now constitute an integral part of any manufacturing system. They help track product quality and process performance. In a typical automated manufacturing system, inspection operations occur several times during a part's sojourn in the factory. Generally it is too expensive to have an inspection operation after every processing step; rather, each inspection is done at the end of some well-defined set of operations. Depending on the outcome of the inspection, the part may be (1) found to be of acceptable quality and forwarded to the next processing step, (2) found to be in need of some retouching and sent back to some machine upstream for rework, or (3) found to be too defective to be reworked and therefore scrapped. It is also not uncommon to have one inspection station performing an inspection on parts coming from more than one machine; in this case, depending on the time required for the inspection, there is a possibility of an inspection facility becoming a bottleneck, and hence, it may be necessary to choose the part scheduling policy at the inspection center judiciously.

Two important problems related to inspection are:

- How many inspection stations to use
- Where to locate the inspection stations

There is much literature on the above two topics. For a review and references, see Viswanadham, Sharma, and Taneja. In both of these problems, one seeks to minimize the expected total cost per unit produced. The total cost includes some or all of the following: inspection cost, diagnosis and repair cost, loss resulting from removal of a unit perceived to be non-conforming less any applicable salvage value, and penalty associated with shipping a nonconforming unit. Both exact and approximate methods have been used to solve these problems. The exact methods used include dynamic programming and integer programming methods. The approximation techniques yield a nearly optimal solution with considerably less computational effort. These include genetic algorithms and simulated annealing.

A few researchers have addressed the problem of performance analysis in the presence of inspections. These works include: Seidmann, Schweitzer, and Nof; Schweitzer and Seidmann; Saboo, Wang, and Wilhelm; Wilhelm; and Davis and Kennedy and the recent work of Connors, Feigen, and Yao.

In this paper, the interest is in investigating the effect of inspections in reentrant manufacturing sys-
tems, which are characterized by distinct multiple job visits to workcenters or processing stations. Examples of such manufacturing systems include semiconductor wafer fabrication facilities, thin film lines, circuit card assembly, and systems with rework tasks. Reentrant lines, a type of nontraditional queuing network model proposed by Kumar, are appropriate for modeling such manufacturing systems. This paper extends the reentrant line model to include probabilistic routing and proposes this as a model for reentrant manufacturing systems with inspections. Also developed is an efficient, approximate analytical technique based on mean value analysis (MVA) to compute the mean steady-state cycle time and throughput rate of such models under various scheduling policies. The new model and the analysis technique facilitate

- predicting the mean steady-state cycle time and throughput rates of reentrant manufacturing systems in the presence of inspections, reworking, and scrapping of parts under a wide variety of scheduling policies, and
- comparing alternative ways of locating inspection stations from a cycle time and throughput viewpoint.

There has been a good deal of recent work on the topic of performance of reentrant lines. Kumar has investigated the performance of a wide variety of scheduling policies based on buffer priorities and on due dates, using simulation. Lu, Ramaswamy, and Kumar have investigated the cycle time and throughput performance of a class of scheduling policies called fluctuation smoothing policies. More recently, Narahari and Khan have used an approximate MVA-based analytical method to evaluate the performance of buffer priority based scheduling policies in reentrant lines and the effect of high-priority jobs or hot lots on the cycle time and throughput of other jobs in a reentrant line. Their work does not take into account the effects of inspections. Connors, Feigin, and Yao have presented an open queueing network model designed for rapid performance analysis of semiconductor manufacturing facilities. They assume first-come first-serve (FCFS) scheduling at all nodes of the queuing network and use a decomposition approach to analyze the model. Their model does capture reworking and scrapping of jobs and different types of failure events but does not model scheduling policies other than FCFS. Many other recent works have addressed the issue of performance analysis by emphasizing performance bounds rather than mean values of performance measures. Also, none of these works consider the effects of inspections.

The next section of the paper presents a model for reentrant manufacturing systems with inspections, where inspection times are negligible, and outlines an analytical methodology based on MVA to compute mean steady-state cycle time under various scheduling policies. Numerical results obtained for a four-machine, 13-buffer example include simulation results to verify the accuracy of the performance estimation of the analytical method proposed. The third section compares the performance of alternate ways of locating inspection stations in reentrant lines using the analytical method proposed. Next is explicitly modeled the contention for inspection stations, and the effects of the resulting congestion are investigated by considering a realistic model of a wafer fab.

**A Model and an Analysis Methodology**

As stated earlier, the proposed model is based on reentrant lines. The model is extended to include probabilistic routing.

**Reentrant Lines with Probabilistic Routing**

A reentrant line can be described as follows. There is a set of service centers \( \{1, 2, \ldots, m\} \). Service center \( i \in \{1, 2, \ldots, m\} \) has \( n_i \) logical or physical buffers, \( b_{i1}, b_{i2}, \ldots, b_{in_i} \). For \( j \in \{1, 2, \ldots, n_i\} \), the buffer \( b_j \) contains parts visiting service center \( i \) for the \( j \)th time (call it stage \((i,j)\) of service). A part visits these buffers in a given sequence and a given service center is typically visited several times in the route of a part. Figure 1 shows a reentrant line with two stations and four buffers, while Figure 2 shows a reentrant line with four service centers and 13 buffers. Parts enter the system at buffer \( b_{i1} \) and visit the buffers at various centers according to a deterministic route as shown. A part visits these buffers in a given sequence, and a given service center is typically visited several times in the route of a part. Figure 1 shows a reentrant line with two stations and four buffers, while Figure 2 shows a reentrant line with four service centers and 13 buffers. Parts enter the system at buffer \( b_{i1} \) and visit the buffers at various centers according to a deterministic route as shown. In Figure 1, finished parts exit after \( b_{22} \), whereas in the case of Figure 2, finished parts emerge after \( b_{44} \). The dotted lines in Figure 2 indicate that a fresh part enters buffer \( b_{i1} \) immediately after a finished part leaves the system, following a constant work-in-process input release policy.
When there is an inspection at the end of a particular stage of processing, it is reasonable to assume three possibilities: accept, reject, or rework. A part that is accepted will queue up for the next stage of processing in the deterministic route. A part that is rejected will be scrapped and will disappear from the system. In this paper, however, it is assumed that a scrapped part is replaced by a fresh, raw part to maintain the work-in-process constant. A part that needs rework may need to be routed to any of the earlier stages of processing. The following assumptions are made regarding parts that need to go for reworking:

1. A part may have to go to any earlier stage, say stage $(i,j)$, for reworking. Such a part is indistinguishable from the one going through stage $(i,j)$ for the first time. This implies that the part going for reworking to earlier stage $(i,j)$ will go through all the stages from $(i,j)$ to the current one.
2. Rework parts have processing requirements identical to original processing times.
3. Parts that go for reworking to a particular stage join the tail of the queue at that stage. This is consistent with an assumption made later that parts within a buffer (that is, stage) are scheduled on a FCFS basis.

After each stage of processing, note that a part may advance to the next stage (if it passes inspection), return to the same stage (rework at the same stage), go back to any previous stage (rework from a previous stage), or get rejected. The probabilities of each of these events are assumed to be known for all stages of processing. An important assumption is that the inspection process is instantaneous. This assumption will be relaxed in a later section.

The inspections, reworking, and rejections detailed above can be described by a reentrant line with a Markovian routing matrix, $P$, where the entries indicate the probability of going from a given stage to any other stage. The matrix $P$ has $b$ rows and $(b + 1)$ columns, where $b$ is the total number of processing stages (or buffers) in the system. Each row of $P$ corresponds to a distinct buffer in the system, and a typical entry gives the probability that a part, after completing service at some stage, will go next to another designated stage. The last entry in each row, except the last row, gives the probability that a part is scrapped after completing that stage of processing. The last entry of the last row is the probability of a finished part emerging from the system.

**Analysis Methodology**

The analysis methodology uses ideas from the well-known mean value analysis (MVA) technique applied in an approximate way to nonproduct form queuing networks. In an earlier paper, Narahari and Khan have already presented an analysis technique for reentrant lines with strictly deterministic routing based on MVA. Here, this methodology is extended to include inspections and, hence, probabilistic routing. The accuracy of the performance estimates of the extended methodology is established by comparing with extensive simulation analysis.

MVA yields expressions for mean values of performance measures such as steady-state queue lengths, delays, and throughputs. Two versions of MVA exist, namely the exact MVA for product form queuing networks and the approximate MVA for nonproduct form networks. Exact MVA is based on the arrival theorem, which states that, in the steady state of a closed product form network with population $k$, the distribution of the network state seen by a job arriving at any node in the network is the same as the distribution of the network state a random observer would see with $(k - 1)$ jobs circulating in the network.

In the literature, several extensions have been proposed to MVA to account for nonproduct form fea-
tures. Of special interest here are the MVA extensions for handling priority scheduling. Most of these approaches consider only preemptive priorities.

The MVA extension proposed in this paper is unique in that it takes into account in a natural way the probabilistic routing of parts in a reentrant line with inspections and multiple job visits to the same workcenter. Also, a special feature of this methodology is that it explicitly models any buffer priority based scheduling policy that may be followed at different service centers; that is, when center $i$ finishes processing a part, it selects the next part for processing from among the buffers $b_{i1}, b_{i2}, \ldots, b_{in}$ in a fixed priority order that is independent of the state of the system. Examples of buffer priority policies in reentrant lines include: FBFS (first-buffer first-serve), where parts visiting a center on a particular visit get priority over parts visiting the center on later visits, and LBFS (last-buffer first-serve), which is the reverse of FBFS. The analysis assumes that the priorities accorded are nonpreemptive and that parts in any given buffer are processed in FCFS fashion.

**Assumptions and Notation**

The formulation of MVA equations is illustrated by assuming LBFS scheduling policy. It is assumed that each processing center has exactly one machine and that the processing time of a job visiting center $i$ on its $j$th visit is an independent exponentially distributed random variable with rate $\mu_i$. In LBFS scheduling policy, parts visiting center $i$ for the $j$th time get priority over parts visiting the center on the $r$th time, where $r = 1, \ldots, j-1$. For example, in center 2 of Figure 2, buffer $b_{25}$ would get priority over $b_{24}, b_{23}, b_{22},$ and $b_{21}$; buffer $b_{24}$ would get priority over $b_{23}, b_{22},$ and $b_{21}$; and so on.

To apply MVA, it is assumed that the reentrant line is a closed queuing network. This assumption is valid if the input release policy is a fixed work-in-process policy (a fresh job is released into the network as soon as a finished job leaves the system). Also, because of inspections, some parts might be rejected at intermediate stages, and this will reduce the number of jobs in the system. To keep the number of jobs in the system constant, it is assumed that a rejected part is immediately replaced by a fresh part that enters the first buffer in the system. Let $N$ be the total number of jobs in the system. The following indices are used: $i$ denotes a processing center, $j$ denotes a buffer at a given processing center, and $k$ denotes a current job population and has the range $1, \ldots, N$. Let stage $(i,j)$ correspond to either the waiting or the processing of a job visiting center $i$ for the $j$th time.

Let the performance measures of the network be denoted as follows:

- $L_y(k)$: mean steady-state number of jobs in stage $(i,j)$ when the network has $k$ jobs
- $W_y(k)$: mean steady-state delay for jobs in stage $(i,j)$ (mean waiting time in buffer $b_y$ + mean processing time)
- $\lambda(k)$: mean steady-state throughput rate of jobs when the network has $k$ jobs

If $W(k)$ denotes the mean total delay (also called mean cycle time) in the entire network in the steady state, then

$$W(k) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \nu_y W_y(k)$$

where $\nu_y$ is the mean number of times a part visits stage $(i,j)$ during its sojourn in the network. It is immediately noted that $\nu_y = 1$ for all $i$ and for all $j$ if there are no inspections in the reentrant line (as a part visits a given buffer exactly once). If there are inspections, $\nu_y$'s from the routing probability matrix can be computed in the standard way as for product form queuing networks.

Using MVA, $W(N)$ and $\lambda(N)$ are computed in a recursive way.

**MVA Equations**

A distinction is made between external and internal buffers. A buffer $b_y$ is called external if the buffer feeding $b_y$ is connected to a center different from center $i$, and buffer $b_y$ is called internal if the buffer feeding $b_y$ is connected to center $i$ itself. This paper considers only the case where all the buffers are external (as in Figures 1 and 2). The case of internal buffers can be handled on the lines detailed in Narahari and Khan.

The calculation of $W(N)$ and $\lambda(N)$ is considered. It would be helpful to consider the scenario a job
would see upon its arrival at a certain buffer of a machine and the sequence of events that occur while it is waiting there.

When a job (called a distinguished job) arrives at a buffer, say \( b_y \), it sees a certain number of jobs in various buffers in the system. The ordered set of these integers forms the state of the system at the arrival instant of the job. Let \( S \) be the set of jobs currently at center \( i \) and having higher priority than the distinguished job. Note that \( S \) includes not only jobs that are ahead of the distinguished job in \( b_y \) but also all jobs in all buffers having higher priority than \( b_y \). The distinguished job must first wait until all jobs in \( S \) are serviced and leave center \( i \). Also, it must wait for the service completion of those jobs that arrive in higher priority buffers during its wait in buffer \( b_y \). And finally, it has to be processed before it enters the next buffer.

Hence, the mean total waiting time of a job at any buffer \( b_y \) is seen as the sum of three components, called Term 1, Term 2, and Term 3, defined as follows:

- **Term 1**: mean total time until all jobs in set \( S \) are serviced and leave center \( i \)
- **Term 2**: mean total time required to process all higher priority jobs that arrive during the stay of the distinguished job in the queue at \( b_y \).
- **Term 3**: mean processing time of the distinguished part itself.

Terms 1, 2, and 3 may be computed by presuming that the arrival theorem is valid in the given network. In fact, the arrival theorem is not valid for this network because the network is not product form. However, because only an approximate analysis is sought, it is first blindly assumed that the arrival theorem is valid for this network, and the accuracy of the approximation is verified using detailed simulation results.

**Computation of Term 1.** Consider the buffer \( b_y \). By assuming the arrival theorem to be valid for this network, an arriving job would see \( L_y(k-1) \) jobs in the buffers \( b_{y,0} \), where \( t = 1, 2, ..., n_i \). Because LBFS scheduling policy is being used, the arriving job needs only to wait for the processing of jobs ahead of it (that is, with higher priorities) in buffers \( b_{y,0} \), where \( t = j, j+1, ..., n_i \). Thus

\[
\text{Term 1} = \sum_{t=j}^{n_i} \frac{L_y(k-1)}{\mu_y}
\]

**Computation of Term 2.** The mean waiting time (excluding processing time) of a job in buffer \( b_y \) is

\[
W_y(k) = \frac{1}{\mu_y}
\]

During this waiting, parts may arrive into higher priority buffers at center \( i \). Term 2 is the mean total time required to process all such parts. Because all the buffers in the model are such that parts may arrive into it only from a buffer in another station, parts may arrive into any of the higher priority buffers (from other machines) while job \( i \) waits in buffer \( b_y \). Now, blindly assuming the arrival theorem to be true for this network, \( \lambda(k-1) \) can be taken as the rate at which the jobs are flowing in the network, and therefore

\[
\text{Term 2} = (W_y(k) - \frac{1}{\mu_y})\lambda(k-1) \left( \sum_{r=j+1}^{n_i} \frac{1}{\mu_r} \right)
\]

**Computation of Term 3.** The mean processing time required for the service of the distinguished part itself is \( \frac{1}{\mu_y} \). Thus Term 3 = \( \frac{1}{\mu_y} \).

The total expected waiting time \( W_y(k) \) is now given by

\[
W_y(k) = \text{Term 1} + \text{Term 2} + \text{Term 3}
\]

Now, using Eq. (1), \( W(k) \) can be computed.

Applying Little’s Law\(^3\) for the population of jobs in the network, the following is obtained:

\[
\lambda(k) = \frac{k}{W(k)}
\]

Little’s Law can again be used to obtain

\[
L_y(k) = \lambda(k)W_y(k)
\]

Using the initial conditions

\[
L_y(0) = 0; \quad i = 1, ..., m \quad j = 1, ..., n_i
\]

\[
\lambda(0) = 0
\]

and the recurrence relations defined by Eqs. (4) to (6), and the initial values Eqs. (7) and (8), \( W_y(k), L_y(k) \), and \( \lambda(k) \) for \( k = 1, 2, ..., N \) can be computed. Thus, \( W(N) \) and \( \lambda(N) \) can be computed.
Complexity of Methodology

For a reentrant line with a total of \( b \) buffers and having population \( N \), the algorithm involves \( N \) iterations. Each iterative step involves computations for each of \( b \) buffers. Hence, the worst-case time complexity of the algorithm is \( O(Nb) \), and thus the algorithm complexity is polynomial in the population of the network and the total number of buffers. Note that the number of buffers is related linearly to the number of processing centers.

As each iteration needs only the values of mean queue lengths and mean throughput rate computed in the previous iteration, the space complexity of the algorithm is \( O(b) \).

If when using simulation there is a desire to calculate the performance parameters for a set of populations, a simulation must run for each value of the population. On the other hand, the proposed algorithm need only be executed exactly once, with the maximum value, say \( N \), of the population of interest. The performance values for all smaller populations \( 1, 2, \ldots, N - 1 \) are automatically produced in the intermediate iterations.

Numerical Example

Consider the reentrant line shown in Figure 2. The service times for all buffers at a given machine center are assumed to be independent, identically distributed exponential random variables. Let \( \frac{1}{\mu_i} \) be the mean service time at each buffer at machine center \( i \). In the analytical and simulation experiments, it is assumed that

\[
\begin{align*}
\frac{1}{\mu_1} &= 0.5 \text{ hr} \\
\frac{1}{\mu_2} &= 0.1 \text{ hr} \\
\frac{1}{\mu_3} &= 0.5 \text{ hr} \\
\frac{1}{\mu_4} &= 0.5 \text{ hr}
\end{align*}
\]

In this reentrant line, machine center 3 is the bottleneck station because it has the maximum service demand among all stations.

Assume that there is an inspection after each processing stage. After inspection at each stage, let the probability of rejection be 0.05 and the probability of going for rework to any of the earlier stages be 0.05. Using the proposed method, and also simulation, the mean cycle time and mean throughput rate of accepted parts for populations ranging from 1 to 35 are computed. The simulations were carried out using a SIMSCRIPT II.5 package on an Intel 80486-DX based machine. A single long run was used to compute the steady-state performance measures, after deleting the transients. The initial transient period in the simulations was determined by making several pilot runs. Also, standard statistical testing was carried out to obtain a 0.05 level for all the simulation experiments. Figures 3 and 4 provide a graphical representation of the numerical results. It can be seen from the graphs that there is close agreement between the results obtained by the proposed method and those from the simulation model. The maximum discrepancy between the simulation and analytical results is about 4% in the case of mean cycle time and about 7% in the case of mean throughput rate.
Comparing Alternative Ways of Locating Inspection Stations

An important problem related to inspection is to determine the optimal number and location of inspection stations in a given manufacturing system. This section addresses the problem of locating a given number of inspection stations and evaluates different alternatives from a cycle time and throughput viewpoint. The methodology is illustrated with an example. Consider the reentrant line of Figure 1, which has four stages, (1,1), (2,1), (1,2), and (2,2). Denote these stages as stage 1, stage 2, stage 3, and stage 4, respectively. Assume that an inspection at the end of stage 4 is always required. For i = 1, 2, 3, 4, define variable $x_i = 1$ if there is inspection at the end of stage i and $x_i = 0$ otherwise. Call $(x_1, x_2, x_3, x_4)$ the inspection location vector. If only one inspection is made, this vector can only take the value $(0,0,0,1)$. If two inspections made, the possible vectors are $(1,0,0,1), (0,1,0,1), (0,0,1,1)$. In the case of three inspections, the possible values are $(1,1,0,1), (1,0,1,1), (0,1,1,1)$. Finally, for four inspections, the only possible vector is $(1,1,1,1)$. Thus, there are eight possible inspection location vectors.

It is assumed that a defect introduced in a given stage does not cause additional defects at succeeding stages. To compare the alternative locations, accept, reject, and rework probabilities are needed in each case. Consider the inspection location vector $(1,1,1,1)$. For $i = 1, 2, 3, 4$, define

$$ r_i = \text{probability of rejection of a part after stage } i $$

$$ q_{ij} = \text{probability that a part goes for reworking to stage } j \text{ after stage } i $$

Note that $q_{ij} = 0$ for all $j > i$. Define $s_i$ as the probability that a part passes the inspection after stage $i$. It is easy to see that

$$ s_i = 1 - r_i - \sum_{j=1}^{i} q_{ij} $$

Assuming that the probabilities $r_i$ and $q_{ij}$ are known for all four stages in the case of the vector $(1,1,1,1)$, these probabilities can be computed for any of the other seven vectors. For example, consider the vector $(0,0,0,1)$. Because there is no inspection at the end of the first three stages, the probabilities only need to be computed for stage 4.

Let these probabilities be $r_4', q_{41}', q_{42}', q_{43}', q_{44}'$, and $s_4'$. It is easy to see that

$$ q_{41}' = s_1 s_2 s_3 q_{41} + s_1 s_2 q_{31} + s_1 q_{21} + q_{11} $$

$$ q_{42}' = s_1 s_2 q_{42} + s_1 q_{32} + q_{22} $$

$$ q_{43}' = s_1 q_{43} + q_{33} $$

$$ q_{44}' = q_{44} $$

$$ r_4' = r_4 $$

$$ s_4' = 1 - r_4 - \sum_{j=4}^{4} q_{4j}' $$

Similarly, these probabilities for all possible location vectors can be computed.

The analysis methodology of the second section can be used for computing the mean steady-state cycle time and throughput rate of accepted, finished parts. Figure 5 shows the mean steady-state throughput rates for all eight inspection location vectors. The mean processing time at each buffer in center 1 is assumed as one unit, whereas that at each buffer in center 2 is assumed to be two units.

Note from Figure 5 that the maximum throughput rate is obtained in the case of $(1,1,1,1)$, that is, complete inspection, and the minimum throughput rate is in the case of $(0,0,0,1)$, that is, no intermediate inspection. This is easy to see because, in the latter case, parts can only get rejected after the last stage. It is also interesting to note that the vectors $(0,0,1,1)$,
(0,1,1,1), and (1,0,1,1) lead to very nearly the same throughput rates. This is significant because two inspection stations located at stage 3 and stage 4 give the same performance as three inspection stations located according to (0,1,1,1) or (1,0,1,1). Also, the vector (0,0,1,1) outperforms the vector (1,1,0,1) in spite of the fact that the former has one less inspection station. This shows that a small number of strategically located inspection stations can perform better than a larger number of poorly located inspection stations. However, it is noted that this conclusion is specific to this example and may not be true in general.

Thus, the proposed technique can be conveniently used to study the comparative merits of different location schemes. It is believed that a study of some representative reentrant lines will lead to a set of heuristic guidelines for choosing the optimal locations for inspection stations. In this case, the location of inspections seeks to minimize cycle time and maximize throughput rate. The model can be extended to include a very general cost criterion as well.

**Modeling Contention for Inspection Stations**

In the analysis methodology presented earlier, it was assumed that the inspection process was instantaneous, which is true in some manufacturing processes but not in general. In this section, this assumption is relaxed by modeling inspection operations that take nonzero time. Here, inspection stations are explicitly modeled like any other processing center except that, after processing (inspection) at each of these stations, the next stage that a part visits depends on the outcome of the inspection operation.

**Part Scheduling at Inspection Centers**

Scheduling of parts at inspection stations should be consistent with the scheduling policy being followed in the plant. For example, if LBFS policy is being followed in the plant, the part scheduling policy at the inspection stations should be so chosen to ensure that LBFS priority ordering is preserved. This is straightforward in cases where each inspection station handles parts coming from only one machine. For example, consider the reentrant line of Figure 6, where center 4 is an inspection station. In this case, all the inputs to center 4 are from center 3.

*Figure 7* shows a reentrant line with four stations (1, 2, 3, and 4) and two inspection stations (5 and 6). Center 5 inspects parts coming from stations 1 and 2, whereas center 6 inspects parts coming from stations 3 and 4, and it is assumed that an inspection takes place after each operation.

**Accuracy of the Proposed Method**

To verify the accuracy of performance estimates obtained using the proposed approximate analytical method, the performance measures for the reentrant line shown in *Figure 6* are evaluated and compared with the results obtained from a simulation model. The simulations were carried out using SIMSCRIPT II.5 on an Intel 80486-DX based machine. A single long run was used to compute the steady-state performance measures. The initial transient period was determined by making an adequate number of pilot runs. Statistics were collected after removing the initial transients. Also, statistical tests were conducted to obtain a 0.05 level of significance for all the experiments.

The system consists of three processing stations and one inspection station. An inspection takes place after

---

**Figure 6**
A Three-Machine, Single Inspection Station Reentrant Line
every three processing steps. The following values of mean processing and inspection times are assumed:

\[
\frac{1}{\mu_{11}} = \frac{1}{\mu_{12}} = \frac{1}{\mu_{13}} = 1
\]
\[
\frac{1}{\mu_{21}} = \frac{1}{\mu_{22}} = \frac{1}{\mu_{23}} = 2
\]
\[
\frac{1}{\mu_{31}} = \frac{1}{\mu_{32}} = \frac{1}{\mu_{33}} = 3
\]
\[
\frac{1}{\mu_{41}} = \frac{1}{\mu_{42}} = \frac{1}{\mu_{43}} = 2
\]

The following rejection probabilities are also assumed:

\[ r_{41} = 0.02 \quad r_{42} = 0.02 \quad r_{43} = 0.01 \]

Rework probabilities assumed are the following:

\[ q_{41,11} = 0.01 \quad q_{41,21} = 0.01 \quad q_{41,31} = 0.01 \]
\[ q_{42,12} = 0.02 \quad q_{42,22} = 0.02 \quad q_{42,32} = 0.02 \]
\[ q_{43,13} = 0.01 \quad q_{43,23} = 0.02 \quad q_{43,33} = 0.03 \]

With these parameters, a comparison of simulated and analytically predicted values of throughput rate is shown in Figure 8. As can be seen, the agreement between the two is quite close. Statistical tests have shown the level of significance of the simulation results to be 0.05. The simulations for the example above typically have a run time of 4 to 5 hours, whereas the analytical method only takes a few seconds.

To be useful, any technique for predicting performance of a reentrant line should not only give gross performance measures, such as mean throughput rate and mean cycle time, but also predict finer grain performance measures, such as mean waiting times or mean queue lengths, at individual buffers of the
system or individual machines. These characteristics are quite valuable in identifying system bottlenecks or hot spots. The technique presented here enables predicting these performance measures accurately. As an example, Table 1 gives waiting times at a designated few buffers of the system of Figure 6. Both analytical and simulation values are shown, and good agreement between these can be seen.

**Table 1**

<table>
<thead>
<tr>
<th>Population</th>
<th>( w_{11} )</th>
<th>( w_{32} )</th>
<th>( w_{41} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td>MVA</td>
<td>SIM</td>
<td>MVA</td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>1.47</td>
</tr>
<tr>
<td>2</td>
<td>1.20</td>
<td>1.21</td>
<td>1.94</td>
</tr>
<tr>
<td>4</td>
<td>1.53</td>
<td>1.62</td>
<td>2.71</td>
</tr>
<tr>
<td>6</td>
<td>1.75</td>
<td>1.87</td>
<td>3.22</td>
</tr>
<tr>
<td>8</td>
<td>1.93</td>
<td>1.99</td>
<td>3.54</td>
</tr>
<tr>
<td>10</td>
<td>1.98</td>
<td>2.05</td>
<td>3.81</td>
</tr>
<tr>
<td>12</td>
<td>2.05</td>
<td>2.08</td>
<td>3.93</td>
</tr>
<tr>
<td>14</td>
<td>2.10</td>
<td>2.11</td>
<td>3.96</td>
</tr>
<tr>
<td>16</td>
<td>2.16</td>
<td>2.12</td>
<td>4.01</td>
</tr>
<tr>
<td>18</td>
<td>2.12</td>
<td>2.13</td>
<td>4.02</td>
</tr>
<tr>
<td>20</td>
<td>2.12</td>
<td>2.14</td>
<td>4.11</td>
</tr>
<tr>
<td>22</td>
<td>2.12</td>
<td>2.15</td>
<td>4.07</td>
</tr>
<tr>
<td>24</td>
<td>2.15</td>
<td>2.15</td>
<td>4.10</td>
</tr>
</tbody>
</table>

Case Study of Realistic Reentrant Line

*Figure 9* shows a model of an actual semiconductor production facility. This model was considered in Lu and Kumar*4 for the comparison of various fluctuation-smoothing scheduling policies. In Lu and Kumar, there is no inspection facility, and hence, rejection and rework are not modeled. Here is included an inspection station in the model at center 13 (not shown in the diagram). It is assumed that the mean processing times for parts taken from all buffers at a given machine are the same, and the system is analyzed assuming the following parameter values:

\[
\begin{align*}
\frac{1}{\mu_1} &= 0.125, \\
\frac{1}{\mu_2} &= 0.125, \\
\frac{1}{\mu_3} &= 0.250, \\
\frac{1}{\mu_4} &= 1.800, \\
\frac{1}{\mu_5} &= 0.900, \\
\frac{1}{\mu_6} &= 0.600, \\
\frac{1}{\mu_7} &= 1.800, \\
\frac{1}{\mu_8} &= 0.200, \\
\frac{1}{\mu_9} &= 0.600, \\
\frac{1}{\mu_{10}} &= 0.333, \\
\frac{1}{\mu_{11}} &= 0.600, \\
\frac{1}{\mu_{12}} &= 1.250.
\end{align*}
\]

It is also assumed that after every inspection a part is rejected with a probability of 0.01 and is accepted with a probability of 0.95. For rework, it is assumed that a part will return for rework only to those buffers that lie between the last inspection point and the current inspection point. The probability of a part being sent back for rework to any of these buffers is assumed to be \( \frac{0.04}{n} \) where \( n \) is the total number of operations between the last inspection and the current operation.

*Figures 10 and 11* plot the system throughput and cycle time for various populations and inspection speeds. The time required for a single inspection was varied from 0.025 to 0.375 time units, and the popu-
lation was varied from 1 to 100. From Figures 10 and 11 is easily seen that the system throughput is a monotonically nondecreasing function of both inspection speed and population, but once the system reaches saturation (that is, maximum bottleneck capacity is reached), just increasing the system population will not result in increased throughput; rather, it will cause the system cycle time to increase rapidly. In such situations, it would be better to increase inspection speed by using faster inspection machines. But this will also help only to a point, after which the throughput will be governed by the most heavily loaded machine (bottleneck) in the system.

Conclusions
This article proposed reentrant lines with probabilistic routing as a faithful model of reentrant manufacturing systems with inspections. Also proposed is an efficient analysis method for the new model based on MVA. The accuracy of the proposed analysis method has been verified using simulation.

It was seen that the proposed method can be used in arriving at important decisions for economical operation of a reentrant manufacturing system, such as:

- determining the minimum number of inspection centers to ensure a given quality level
- determining the optimal location of a given number of inspection centers to maximize the throughput of quality parts
- determining the lowest possible speeds of the inspection stations that will not hamper the system performance
- computing various performance measures for any given layout of a reentrant manufacturing system, taking into account the effects of inspections and contention for inspection stations

Acknowledgments
This research was supported in part by the Office of Naval Research and the Department of Science and Technology grant N00014-93-1017. The authors also acknowledge the excellent facilities at the Intelligent Systems Laboratory, Department of Computer Science and Automation, Indian Institute of Science.

References


Authors’ Biographies
Y. Narahari is an associate professor in the Department of Computer Science and Automation at the Indian Institute of Science (Bangalore, India). His teaching and research interests are in the areas of performance modeling, deterministic and stochastic scheduling, and object-oriented design and analysis. He has a PhD from the Indian Institute of Science, and his doctoral dissertation on Petri net models of flexible manufacturing systems won him a Gold Medal for the Best Thesis in 1988. He has coauthored, with N. Viswanadham, a critically acclaimed textbook entitled Performance Modeling of Automated Manufacturing Systems (Prentice-Hall, 1992). During 1992, he was a Indo-U.S. visiting scientist at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He has written several widely referenced papers on Petri net models of manufacturing systems, deadlock analysis of manufacturing systems, performance analysis of discrete event systems, and scheduling of semiconductor manufacturing systems.
L.M. Khan received his BE in electronics from the Aligarh Muslim University (Aligarh, India), his MS in computer science from the Indian Institute of Science, and his PhD in computer science and automation from the Indian Institute of Science. He is currently a senior design engineer in the mobile communications division of Hughes Software Systems Limited (New Delhi, India). His current research interests are in the design and analysis of scheduling strategies for queueing models of manufacturing systems and data networks.