ISSUES IN THE DESIGN AND OPERATION OF FLEXIBLE ASSEMBLY SYSTEMS FOR LARGE PRODUCTS: A SIMULATION STUDY

Candace A. Yano

Heungsoon F. Lee

M.M. Srinivasan

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

Technical Report 88-10

July 1988

Revised August 1989

To appear in Journal of Manufacturing Systems

DESIGN AND SCHEDULING OF
FLEXIBLE ASSEMBLY SYSTEMS FOR LARGE PRODUCTS:
A SIMULATION STUDY

Candace Arai Yano
Heungsoon Felix Lee
Mandyam M. Srinivasan

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

July 1988
Revised August 1989

DESIGN AND SEQUENCING OF FLEXIBLE ASSEMBLY SYSTEMS FOR LARGE PRODUCTS: A SIMULATION STUDY

I. INTRODUCTION

This study was motivated by a major automobile manufacturing company which was considering a flexible assembly system as an alternative to a paced assembly line. In a paced assembly line, jobs enter the assembly line at a constant, deterministic rate, usually on a constant speed conveyor, and each workstation receives the same sequence of jobs. On average, each workstation on the assembly line has one *cycle time* to complete the operations required by a vehicle, where the cycle time is defined as the time between successive arrivals of jobs. Paced assembly lines which simultaneously produce a variety of product models tend to have fairly low labor efficiency (e.g., 60% to 80% utilization) because of difficulties in balancing the line and sequencing jobs in such situations.

A flexible assembly system, on the other hand, permits asynchronous transfer of jobs between workstations. These systems typically consist of a number of assembly workstations, each with a human or robot capable of performing a variety of tasks, and a material handling system that permits some degree of routing flexibility among the assembly workstations. The versatility of the assemblers in conjunction with the routing flexibility permits simultaneous assembly of several different products. Here, a job moves from one workstation to the next workstation as soon as it can, and a workstation starts processing as soon as a new job becomes available to it. The flexible assembly system has the potential advantage of a higher labor utilization. This is discussed in the next section, along with other related issues.

The designers of the proposed system at the automobile company were interested in developing some insight about these flexible assembly systems and in determining whether they would be more economical and efficient than traditional paced assembly lines. Thus, our study investigation, the servers within each group are assumed to be identical in terms of capability and speed. Such a group of identical servers will be referred to as a node in our discussion. In queueing parlance, the system has multi-server queues in tandem. A conceptual diagram of the system being considered appears in Figure 1.

Figure 1

The major design decisions to be made are:

1. the number of servers in each group,
2. the size and type of input buffer for each group, and
3. the size of the storage bank for kits to support each group.

Sequencing decisions to be made include:

1. the queue discipline for each group, and
2. the delivery sequence for kits at each storage bank.

We are also interested in determining the effect of the variability of the processing times (across jobs) at a group of workstations on the throughput of the system and the amount of excess material handling effort required to ensure proper mating of jobs with component parts.

The remainder of this paper is organized in the following way. In the next section we describe the motivation for and concept of the proposed system and explain in some detail why it might be preferred to more traditional paced assembly lines. This is followed by a brief review of related literature. We then describe the context within which the proposed system would operate. The subsequent section contains a discussion of the simulation model of the system, followed by a discussion of experimental results and conclusions.

transfer necessitates buffers between workstations to maintain the high level of labor utilization that is generally desired in automobile assembly plants.

In addition to using asynchronous transfer, it has been suggested that the number of assembly operations assigned to a workstation should be increased. This can contribute to reducing the coefficient of variation of the processing times, as follows. In reality, each operation has a (slightly) random processing time, and each job requires a different set of operations. Thus, except in instances where option selections are highly correlated, one might expect the clustering of larger numbers of assembly operations at a single workstation to have a portfolio effect, thereby moderating the variances of the assembly times. This was indeed the situation in the proposed system, where it was anticipated that, if operations were clustered, the shortest and longest processing time at a workstation would be no more than about 20% from the mean processing time, even accounting for possible rework to be done at the workstation.

As a consequence of clustering more operations at a workstation, it may be necessary to have several (identical) assemblers in parallel performing each cluster of operations to ensure that production requirements are met. This, in turn, makes it possible to consider using interstage buffers since it might be feasible to place buffers between pairs of consecutive *groups* of workstations. These inter-node buffers can contribute towards maintaining high levels of labor utilization and throughput by reducing blocking and starving. On the other hand, introducing interstation buffers between every pair of adjacent workstations in an asynchronous *serial* system is impractical from the standpoint of space and cost considerations in an automobile assembly facility which typically has hundreds of workstations. Thus, by simultaneously considering clustering of operations and asynchronous transfer, it might be possible to achieve high labor and machine utilization without the need for a large investment in buffers.

One other problem that arises in some paced assembly lines is that of coordinating the sequencing of jobs and the associated assembly parts. On the surface, this would appear to be trivial. If, however, the sequence of jobs becomes shuffled because of (offline) repair delays or machine breakdowns, then the sequence of assembly parts needs to be modified. For large

Wilhelm and Ahmadi-Marandi (1982) develop a methodology for describing the operating characteristics of small lot assembly systems with single-server workstations operating in tandem. They assume infinite buffers between workstations, and instantaneous transport time between one workstation and the buffer for the next workstation. A numerical approximation technique is developed for predicting the performance of such a system, assuming both deterministic and random job processing times. The impact of certain scheduling policies that specify both the intervals at which jobs are dispatched into the system, and the due dates for delivery of the finished job, on performance is also studied.

The above studies consider single-server queues. When there are multiple-server queues with variable processing times, a job (call it job B) that begins service after another job (job A) may, in fact, depart before job A. This phenomenon, referred to as overtaking, is studied by Whitt (1984), who defines two measures of overtaking: a) the random number of customers overtaken by an arbitrary customer, and b) the degree of shuffling of the incoming stream of customers on departure. Whitt presents several conjectures about the amount of overtaking in Jackson networks (Jackson 1963), and the impact of overtaking on the waiting times.

Udomkesmalee and Daganzo (1986) examine a single workcenter with multiple servers, having variable processing times. They study the degree to which the jobs get out of sequence, and present an approximate analytic approach to derive formulas that estimate this under the assumption that the processing times follow a normal distribution. These formulas are then applied to estimate the number of buffers needed to ensure that 99% of all jobs can be resequenced to their original positions. The approximations are validated with simulations. In a related paper, Udomkesmalee and Daganzo (1987) consider the impact of the variable processing times on the sequence of jobs leaving a single m-server assembly workcenter where a job is mated with some parts after leaving the workcenter (presumably in preparation for assembly operations at the subsequent workcenter). They present two strategies for dealing with the resequencing problem. One strategy is to use buffers for resequencing the jobs. For this strategy, an estimate of the buffer size is obtained by assuming that the m servers at the installation of the seats and the power train, would not be done within the new configuration because of the complex material handling involved in delivering large and heavy parts to the assembly line, and the specialized equipment involved in installing some of these parts. The flexible assembly system would be preceded by the body and paint shops which, because of technological considerations, would operate using fixed pace conveyors. The painting operations would be followed by some of the trim and chassis operations which would be performed on a fixed pace line. This would be followed by the flexible assembly system. Following the flexible assembly system would be other trim and chassis operations that need to be done on a fixed pace line, and final assembly operations. Thus, one can view the proposed flexible assembly system as being fed by a fixed pace line and feeding another fixed pace line.

This configuration may seem, at first, to be rather unusual. However, there are many instances of flexible assembly operations where jobs are dispatched to the system at a fairly steady rate. For example, in most unidirectional flow systems, it is customary to dispatch a job whenever a machine in the first group of servers in the system becomes free, and for most real systems, this occurs at a relatively steady rate. Moreover, most flexible assembly systems do not operate in isolation, but are fed by processing and fabrication stages that operate in a fairly repetitive way.

V. SIMULATION MODEL

The analysis of the system that is being considered here presents considerable difficulties. Typically, the analytical models assume Poisson arrivals in order to simplify the analysis. Even for the single multiserver node with Poisson arrivals and finite buffers, no exact results are available when service times following a general distribution, although some approximations have been reported (Nozaki and Ross 1978, Miyazawa 1986). When a system of such multiserver queues in tandem is considered, the analysis becomes very difficult. This is because the departure process from each queue, which constitutes the arrival process to the next downstream queue, is no longer Poisson (and this is true even if the service times at each node buffer for the next set of workstations, and one path between a kit storage area and the input buffer for its associated group of servers. (This configuration is one under discussion by the designers of the system.) Thus, if a job arrives at the kit storage area and its parts are not available, it waits for its kit of parts to arrive. If, upon arrival of a job to the kit storage area, the parts are not even in-transit to the kit storage area, it is assumed that an expeditor is dispatched to retrieve the parts.

We model two different service disciplines within the simulation. The first is the first-come-first-served (FCFS) discipline. The second is a priority scheme in which the original input sequence to the system is strictly maintained at each group of servers. This rule is sometimes referred to as "in-line-sequence", or "in sequence" (INSEQ). There are several different ways in which this strict sequencing can be maintained. Next, we describe the one that we consider in our study. Later in the paper we will discuss an alternative rule which lies somewhere between FCFS and strict sequence maintenance.

The INSEQ discipline requires that the sequence of entry into each input buffer be consistent with the original input sequence. This ensures that the order of service is the same as the original sequence and that each group of servers can use a simple FCFS policy. Consequently, a simple (linear) conveyor that permits asynchronous movement can serve as an input buffer, keeping the material handling investment low. (Note that the same configuration can be used with the FCFS policy.) In addition, the kit delivery process is simplified because the sequence of jobs arriving to the kit storage bank is exactly the same as the job sequence. Thus, it is unlikely that a job would need to wait for its kit of parts, even if there were only space for one kit of parts in the storage bank. On the other hand, unless the service times are deterministic, this layout can cause blocking (e.g., while job n+1 waits for job n to be completed), and starving (e.g., while a group of servers waits for one tardy job).

We use two different processing time distributions. The first is exponential, which was selected because we are interested in determining whether this distribution, which is often assumed in order to simplify queueing analyses, is an appropriate approximation for this system.

corresponding kit, the resulting expedited kit is able to access this job directly without accessing the kit buffer first. With this assumption, it is technically impossible for such deadlocks, as described above, to occur.

The simulation model was coded in SIMAN (Pegden 1986) along with some Fortran routines which interfaced with the SIMAN code. In Figures 2 and 3 we provide flowcharts which indicate how a job proceeds from one station to another under the FCFS and INSEQ disciplines, respectively. This should provide the reader a sense of how jobs flow through the system. It is important to note that there is one such flowchart being processed simultaneously for each job in the system and that a similar flowchart must be processed for each stage of processing of a job. Meanwhile, jobs are being introduced at the rate of one per unit time.

Figure 2

Figure 3

VI. EXPERIMENTAL INVESTIGATION

In our investigation, each simulation replication consisted of ten eight-hour shifts (i.e., one week). We started the system empty and idle, and through experimentation found that a warmup period of one shift was more than adequate to eliminate transient effects. Each reported value is based upon the average (or where appropriate, the maximum) of three simulation replications. Although we originally planned for many more replications, we quickly discovered that the variance among the results was extremely small because of the length of each replication. Thus, we did not perform tests of statistical significance, but the general patterns are quite strong.

We performed a series of simulation experiments, with each being based upon previous results. This allowed us to identify key factors early in the investigation, thereby eliminating some of the simulation runs that otherwise would have been required with a full factorial design. We also considered a full factorial design to be impractical in view of the exploratory nature of the study and the fact that we were using hypothetical (but representative) processing time data

Problem Parameters and Factors to be Investigated

In the data obtained from the automotive company, various tasks were divided among 6 nodes, with the workloads being fairly equally balanced. The processing time data appear in Table 1. For the triangular distribution, the minimum and maximum values are equal to 80% of the mean, and 140% of the mean, respectively. This distribution was selected to be consistent with data provided to us. We assumed that the transit time between nodes was 2.0 minutes, that the normal kit delivery time was 2.0 minutes, and that the expedited kit delivery time was 0.5 minutes. The latter two figures might not be achievable in some situations. For this reason, we also used longer delivery times, but the general patterns and qualitative conclusions are not affected by the delivery durations as we explain below.

Table 1

With the data given above, it would have been possible to signal the kit delivery system upon *departure* of the job from the previous node. However, we assumed instead that the kit delivery system at a given node is signaled upon the *start of service* of the job at the previous node. Thus, our only implicit assumption is that the delivery process takes less time than the average processing time at a node, and this is quite typical in practice. Consequently, while larger kit buffers are desirable as the duration of the kit delivery process increases, the role of the kit buffers in accomodating sequence changes due to processing time variability remains the same.

To reduce the number of parameters to be considered, we initially attempted to design a cost-effective system using the FCFS policy, assuming that the service time distributions were triangular. This configuration serves as a baseline from which various effects can be investigated. In designing this system, we implicitly assumed that servers were the most expensive resource, followed by the input buffers, and finally the buffers for kits of parts. This relationship is true in our specific application, but our experience suggests that it is true in most

Experimental Results

Recall that the two primary measures of performance are throughput and number of expedited kit deliveries. The estimated throughput values, the fraction of job arrivals *entering* the system (i.e., not blocked from entering the first input buffer), and the fraction of kit deliveries expedited at nodes 2 and 6 are reported in Table 2 for the FCFS discipline for selected combinations of service time distributions and kit storage buffer size. There are some striking patterns in these results, which we describe below.

Table 2

First, with triangular processing time distributions, the FCFS service discipline permits the system to achieve the desired throughput with as few as six kit buffer spaces, and increasing the kit buffers serves principally to reduce the fraction of kits that must be expedited. Thus, when using this service discipline, one should evaluate the economic tradeoff of capital costs for kit buffer spaces versus the cost of expediting (stock chasers).

Secondly, when the FCFS policy is used with exponential processing time distributions, throughput degrades and in this instance, most of the degradation is *not* due to insufficient kit buffers. Observe from Table 2 that with 24 kit buffer spaces there is no need for kit expediting; thus, in general, jobs need not wait for their respective kits of components. Instead, the throughput degradation is attributable primarily to starving of downstream servers and to insufficient input buffer spaces to deal with the high processing time variances. Thus, in a system having highly variable processing times, it appears that large input buffers are required to maintain the desired throughput, and large kit buffers are required to ensure a reasonable level of kit expediting.

We also observed a strong (but intuitively obvious) interaction between the size of the input buffer and the size of the kit buffer for a node. In particular, systems with the FCFS discipline performed poorly (in terms of both throughput and kit expediting) when the kit buffer was smaller than the input buffer. This occurs because the last few jobs in the input buffer often simulation results on how far ahead of its original position in the sequence each job became at each of the six nodes. In Table 4, we report the largest deviation (i.e, original position minus actual position) over the three simulation runs at nodes 2 and 6.

Table 4

The values in Table 4 provide an indication of how large a resequencing buffer would be required to return the jobs to the original sequence. The column labeled "node 2" lists how far out of sequence the worst case job was at node 2, i.e, after one opportunity for getting out of sequence. If we were to resequence the jobs to return them to their original positions after each node, we would need random access buffers of the size indicated here. The column labeled "node 6" lists how far out of sequence a job was at node 6, in the worst case, given that there is no resequencing at intermediate stages. Thus, the value indicates how large a random access buffer would be necessary if the only resequencing occurs at the end of the FAS and the jobs had to be returned to their original positions in the sequence. (Of course, in the case of exponential service times, it is theoretically possible for the necessary buffer sizes to be infinite.)

In either case above, there is a fair amount of resequencing that must be done. In order to accomplish the resequencing, random access buffers are normally required. Such buffers are much more expensive (for equipment, space, and control software) than a conveyor which serves as a sequential access buffer. Thus, there is a fundamental tradeoff between the capacity to resequence jobs and the capacity to resequence parts. It is also important to point out that resequencing necessitates delaying some jobs, which in turn normally leads to occasional starving of downstream workstations, resulting in decreased throughput. Of course, if an FAS does not feed a fixed pace assembly line and there is no compelling reason to maintain the original sequence of jobs, resequencing is not necessary at all. Further research is needed to quantify the effect of resequencing policies upon throughput.

On the other hand, the INSEQ discipline requires more servers, more input buffer spaces, or both, in order to achieve the desired throughput. We next investigated whether additional input buffer spaces alone would increase the throughput to the target value. We found variances, the issues that are normally perceived as critical (i.e., number of servers at each node and the size of the input buffers) appear to be relatively easy to decide. At the same time, decisions such as the sizes of component part buffers and the existence, location, and sizes of job resequencing buffers may be quite critical, yet fairly difficult to specify.

Our original goal was to identify and to gain insight about major design tradeoffs in flexible assembly systems for automotive assembly. Although there is certainly much more research needed, we believe that this research represents an important step toward this goal.

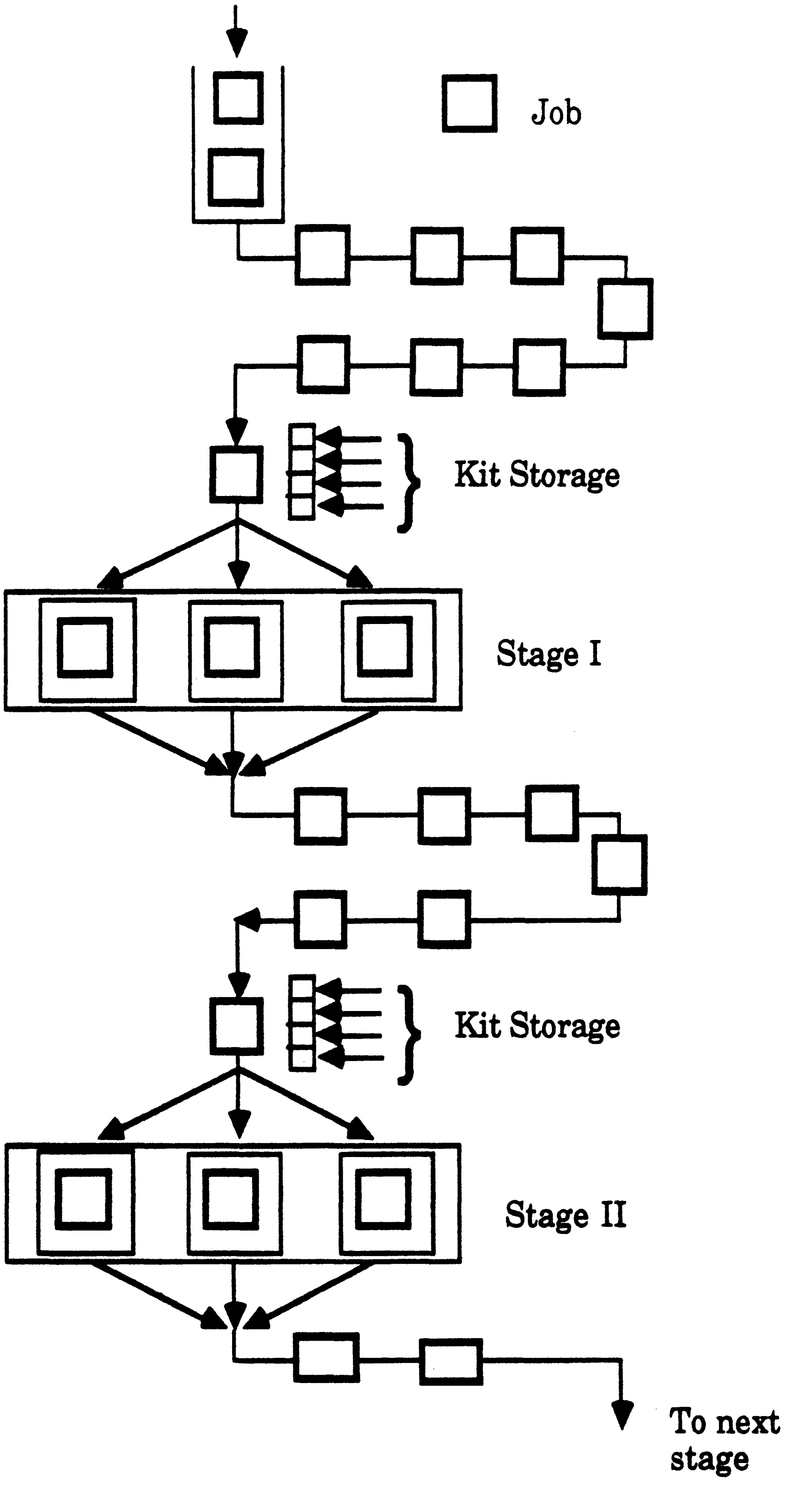

Figure 1: Conceptual diagram of the Flexible Assembly System

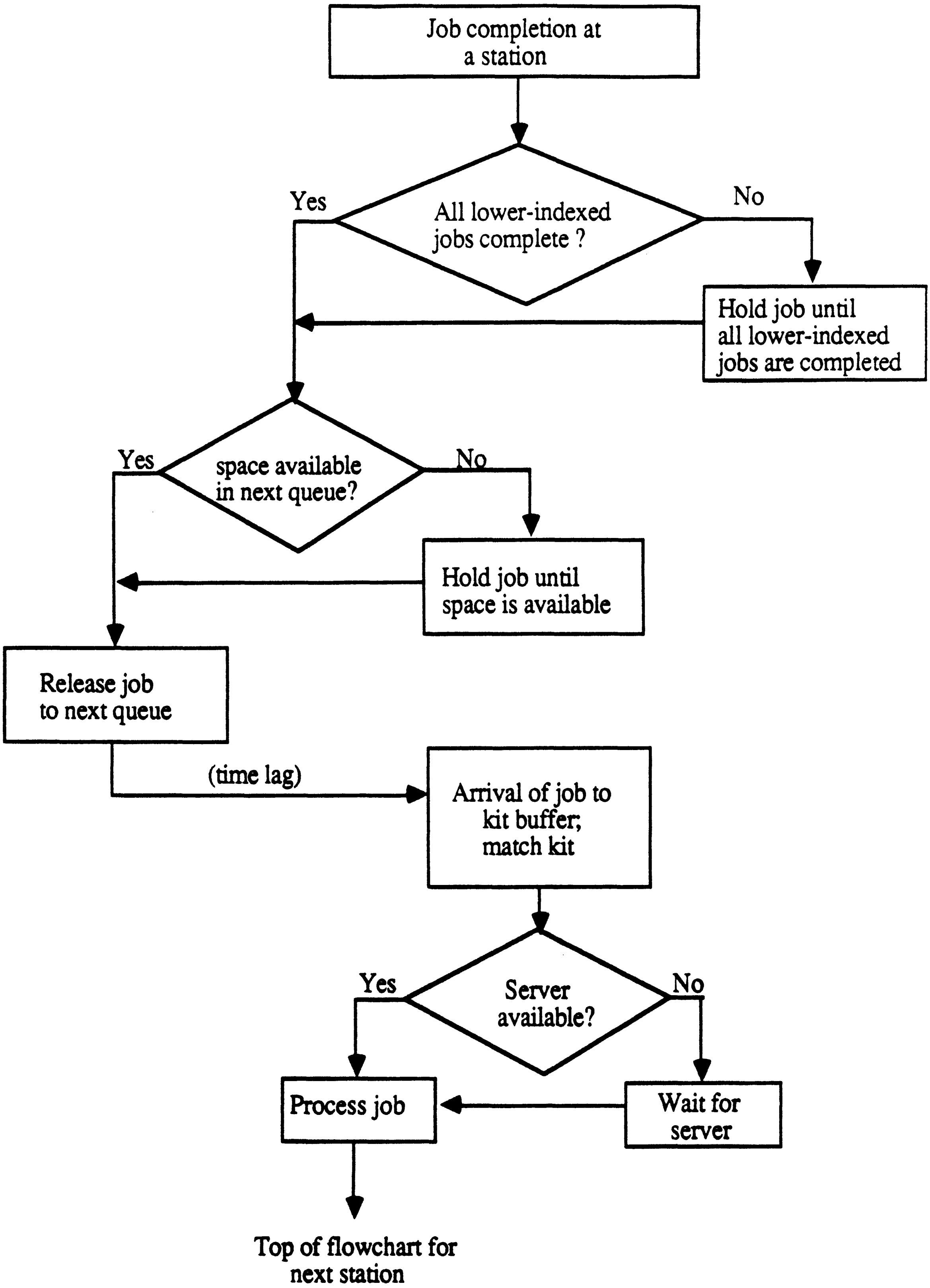
Job completion at a station
All lower-indexed jobs complete ?
Yes
No
Hold job until all lower-indexed jobs are completed
space available in next queue?
Yes
No
Hold job until space is available
Release job to next queue
(time lag)
Arrival of job to kit buffer; match kit
Server available?
Yes
No
Process job
Wait for server
Top of flowchart for next station

Figure 3: Typical flowchart for a job at a station

– INSEQ Discipline

Table 2

Simulation Results: Throughput and Kit Expediting
for FCFS Service Discipline

Service Time Distribution	Kit Buffers	Estimated Throughput (units/time)	Entry Rate (units/time)	% of Kits Expedited (node2)	% of Kits Expedited (node 6)
Triangular	6	0.9986	1.0000	0.14%	0.77%
	10	0.9991	1.0000	0	0
Exponential	10	0.9073	0.9136	32.7%	35.9%
	24	0.9203	0.9251	0	0

Table 4

Maximum "Out-of-Sequence" Deviations for FCFS Discipline
(original position less actual position)

Service Distribution	Kit Buffers	Node 2	Node 6
Triangular	6	7	17
	10	7	17
Exponential	10	16	69
	24	16	74