Pergamon

Omega, Int. J. Mgmt Sci. Vol. 26, No. 3, pp. 381–395, 1998

Printed in Great Britain
0305-0483/98 $19.00 + 0.00

PII: S0305-0483(97)00066-2

JIT Scheduling Rules: a Simulation Evaluation

SIN-HOON HUM

National University of Singapore, Singapore, Republic of Singapore

CHEE-KWONG LEE

National University of Singapore, Singapore, Republic of Singapore

(Received February 1996; accepted after revision October 1997)

Just-In-Time (JIT) production systems capitalize on simplicity and the ability of workers to make decisions in a decentralized manner. In a multiproduct line operating under kanban control, the production worker at a given station must schedule the different jobs awaiting processing using information available locally. In practice, the first-come-first-served (FCFS) rule is commonly used. Recent results reported in the literature indicated that the shortest-processing-time (SPT) rule performed better than the FCFS rule. In this paper, we provide a simulation evaluation of the performance of a number of scheduling rules operating under different JIT production scenarios. Our hypothesis is that there are differences in the relative performance of the scheduling rules under different production scenarios. We differentiate among the JIT scenarios by the extent of setup time reduction already carried out (as indicated by the ratio of setup to processing times), the amount of slack in the system (as measured by the number of kanbans circulating), the extent to which uncertainty has been eliminated (as determined by the stochasticity of processing times), and the complexity of production requirements (as specified by the product-mix in mixed-model assembly). In this way, this paper provides further insights into the performance of scheduling rules operating under different JIT production environments, thereby adding to the scope and depth of research in this particular aspect of JIT production systems.

Key words—Just-In-Time, scheduling, simulation

1. INTRODUCTION

IN A MULTIPRODUCT, multistage line operating under kanban control, the production worker at a given station must schedule the different jobs awaiting processing using information available locally. This is in line with the characteristics of JIT systems in that they capitalize on system simplicity and the ability of line workers to make decisions in a decentralized manner. Hence the line worker will have to decide which of the different kanbans representing containers of different products and queueing at the station needs to be processed first.

In practice, the first-come-first-served (FCFS) rule is commonly used. Recent results reported in the literature [1, 2] indicated that the shortest-processing-time (SPT) rule performed better than the FCFS rule. In these instances, the JIT system studied took the form of an eight-station serial line with no transport delay between processing stations. On the other hand, a subsequent study considered a six-station flow line with a periodic material handling mechanism as opposed to continuous material flow (see Berkley and Kiran [3]). The results reported in this latter study indicated that larger average inventories were associated with the SPT rule when compared to that of the FCFS rule. Similarly, a further subsequent study [4] evaluated a single-card kanban-controlled six-station line with limited queue capacities and reported that the performance of SPT declined relative to FCFS for larger queue capacities and less frequent material handling.

Our hypothesis is that there are differences in the relative performance of the scheduling rules under different JIT production scenarios. In this paper, therefore, we consider a six-station single-card kanban-controlled line and provide a simulation evaluation of the performance of a number of scheduling rules operating under different JIT production scenarios. We differentiate among the JIT scenarios by the extent of setup time reduction already carried out (as indicated by the ratio of setup to processing times), the amount of slack in the system (as measured by the number of kanbans circulating), the extent to which uncertainty has been eliminated (as determined by the stochasticity of processing times), and the complexity of production requirements (as specified by the product-mix in mixed-model assembly). In this way, this paper provides further insights to the performance of scheduling rules operating under different JIT production environments, thereby adding to the scope and depth of the existing research in this particular aspect of JIT production systems.

This paper is therefore organized as follows. In the next section, we provide a review of the existing work in this aspect of JIT scheduling. This is followed by a description of the scheduling rules that will be evaluated in our study, together with details of the simulation model constructed and the parameters that will be varied for the different experiments in representing the various JIT production scenarios. We then report on our simulation results and highlight their implications. We conclude the paper with a summary and some suggestions for further work.

2. RELATED LITERATURE

There is now a substantial body of literature on JIT production systems since the publication of the first paper by Sugimori *et al.* in 1977 on the Toyota Production System [5]. The first comprehensive book written on the overall topic of Japanese production systems is that by Schonberger in 1982, while the first comprehensive book written specifically on the Toyota Production System is that by Monden in 1983 [6, 7]. Since these early literature, there has been a proliferation of JIT studies including both empirical studies on the adoption and implementation of JIT systems (see Refs [8–12] as examples), as well as theoretical studies on the design and performance of JIT manufacturing (see Refs [13–17] as examples). We refer the reader to two fairly recent reviews of the literature on JIT production by Keller and Fouad [18] and by Golhar and Stamm [19]. For a comprehensive overall review of the entire area of JIT manufacturing, we refer the reader to the recent book by Korgaonker [20] which provided a detailed review of the theoretical published work as well.

Despite the subtantial body of literature available on JIT production systems, there has been little work done on the specific aspect of scheduling within the JIT system. While Lummus [21] and Philipoom *et al.* [22] studied sequencing rules in JIT production, their concern was with final assembly sequencing. This, however, is not the focus of our study. Instead, our study here is concerned with the localised choice of scheduling rules at any given production station within the JIT system. In this regard, Lee's work [1] was represents the first published work that explicitly studied the performance of scheduling rules within the JIT environment. Lee provided a simulation analysis of five scheduling rules using an eight-station serial line with no transport delay between stations. The five rules studied were the following: FCFS, SPT, Higher Pull Frequency (HPF), SPT/LATE, and HPF/LATE. The first two rules are well known and widely used. The HPF rule gives priority to the item that is pulled the most by the downstream station as measured by the number of kanbans of that item at the station. The last two rules represent the basic rules with an added check for lateness in the system. A job is termed late if it is not possible to proceed with the processing of a job in a subsequent station due to the inability to pull from the preceding station.

Lee evaluated all five scheduling rules under a specific given scenario. The SPT/LATE rule was found to perform generally better than the other rules in terms of the set of criteria used, namely: the number of jobs actually drawn from the line, the mean job tardiness, the mean job queue time, process utilization, cumulative setup time to operations time ratio and the output kanban inventory level. Lee then further compared the performance of the SPT/LATE rule against that of the FCFS and HPF/LATE rules as the average daily pull demand was varied. Again, the SPT/LATE rule performed better. Lee then finally evaluated the SPT/LATE rule further by analyzing its performance as the production kanban size and minimum output kanban level were varied. Lee also tested the performance of the SPT/LATE rule as the job mix is varied. In this way, Lee's study was focused primarily on the effectiveness of the SPT/LATE rule. The performance of the other rules were not evaluated under different production scenarios.

Lee and Seah [2] added to the work of Lee [1] by studying the performance of two of the scheduling rules (FCFS and SPT/LATE) as changes were made to the processing times. Lee and Seah apparently used the same eight-station serial line model as in Lee [1]. They tested the two rules using the following distributions for processing times: the negative exponential, the constant processing time, and the normal distribution with coefficients of variation of 0.2 and 0.4. Their results indicated that the two rules performed better when processing times were constant or normally distributed; performance deteriorated when processing times were negative-exponentially distributed. In the latter case, the SPT/LATE rule performed significantly better than the FCFS rule. Lee and Seah then evaluated the SPT/LATE rule as setup times were varied. In this way, as in the earlier work of Lee, they apparently again focused their efforts primarily on the performance of the SPT/LATE rule.

Berkley and Kiran [3] also studied the effectiveness of scheduling rules in JIT production systems [3]. Their focus, however, was on the performance of FCFS, SPT, and the related SPT/LATE and FCFS/SPT rules. Also, their simulation model is based on a six-station dual-card Kanban-controlled flow line with a periodic material handling mechanism as opposed to continuous material flow. In this way, Berkley and Kiran's study also seek to investigate the effects of the material movement period on the performance of the rules.

Using performance measures relating to inventory (the average input, output and total work-in-process inventories) and the waiting time of finished-goods withdrawal kanbans, their results indicated the generally weaker performance of the SPT rule vis-a-vis the FCFS rule. The SPT rule not only involved a larger average inventory level but also a longer average finished-goods withdrawal kanban waiting time. Also, the performance of the rules were clearly affected by the material withdrawal period.

Berkley [4] extended the work of Berkley and Kiran by focusing on the SPT and FCFS rules and evaluating their effectiveness in the presence of queue capacity and the periodic material handling mechanism. Berkley, however, used a simulation model based on a six-station single-card kanban-controlled line. Berkley reported results that indicated the deterioration of the performance of the SPT rule relative to the FCFS rule for larger queue capacities and less frequent material handling.

Berkley [23] apparently also used the same simulation model to evaluate the effectiveness of the SPT and FCFS rules under varying levels of station input buffer capacity. Berkley's motivation here is derived from the commonly held objective of buffer capacity reduction for progressive JIT systems. The simulation results reported by Berkley showed that the SPT rule performed better when input buffer capacities are small (and processing times are exponential), while the FCFS rule did better when capabilities are large (and processing times are normal).

More recently, Chao *et al.* [24] introduced their K-priority problem which is described as the scheduling problem at a single server station (in a non-serial production line) linked with several different downstream stations via separate kanban loops. While Chao *et al.* stated that they were unable to find any prior work in which the K-priority problem had been specifically addressed, their problem is actually similar to the kanban scheduling problem highlighted in the literature reviewed here. In any case, Chao *et al.* [24] proposed a new sequencing rule referred to as the K-rule by using Markovian decision process and dynamic programming. While their computational tests showed the superiority of the K-rule versus the FCFS and the LQ (longest queue) rules, the computational effort required in the implementation of the K-rule makes it impracticable. This is especially so when we seek to consider only scheduling rules that can be readily implemented using only locally available data.

Also recently, Yavuz and Satir [25] provided a comprehensive simulation study of the mixed model kanban controlled JIT manufacturing line involving the use of seven different performance measures and nine different experimental factors, including that of sequencing rules. In their conclusion, with regard to sequencing rules, they stated that in general, the SPT rule showed superior performance, especially in more restricted operational environments. They compared the SPT rule with the FCFS and HPF rules.

There is therefore no clear dominance of one scheduling rule over the other with regards to the performance of SPT versus FCFS. Our hypothesis is that there are differences in the relative performance of the scheduling rules under different JIT production scenarios. Therefore in this paper, we evaluated the performance of a given set of scheduling rules under different operating environments of the JIT system, seeking to yield results that may suggest the relative strength of each rule in a given type of JIT environment. Thus while we included the FCFS and SPT rules, we also considered two other kanban-related rules and evaluated all these rules using a common set of JIT operating scenarios. Our objective is to evaluate the relative effectiveness of the rules for different JIT operating scenarios as represented by the extent of setup time reduction already carried out, the amount of slack in the system, the extent to which JIT practices have led to the elimination of uncertainty in processing times, and the complexity of production requirements. In this way, this paper seeks to add to both the scope and depth of the existing research in this particular aspect of JIT scheduling.

3. MODEL DESCRIPTION

In this section, we describe the scheduling rules that we evaluated, the simulation model that we constructed, and the parameters that we varied to represent the different types of JIT scenarios.

3.1. Scheduling rules

We studied the performance of these four scheduling rules: the first-come-first-served (FCFS) rule, the shortest processing time (SPT) rule, the number of kanbans (NKB) rule and the ratio of kanban (RKB) rule. The first two rules are the commonly used and studied rules in industries and in the literature. The NKB rule, equivalent to Lee's HPF rule in Ref. [1], gives priority to producing the type of parts that has the greatest number of kanbans waiting at the workstation. This rule therefore takes into account the frequency of demand for each part type as represented by the kanbans awaiting processing at the station. The ratio of kanban (RKB) rule has the advantage of considering the likelihood of parts shortages. This ratio is based on the number of kanbans of a particular part type awaiting processing relative to the total number of kanbans of that part type that the system started out with between that station and its pulling station. The RKB rule gives priority to the part type that has the largest ratio since the risk of starving the downstream pulling station of this part type is apparently the greatest.

3.2. Model structure

For our study, we developed a six-station, multiproduct, single-card kanban-controlled serial production line. We limited the number of stations to six, inclusive of the final assembly stage, while the number of products would be varied between two and five. The choice of six stations was made since previous reported research have used models with a relatively similar number of stations. More importantly, by running our model prior to the main experiments, we found that six stages was sufficient to highlight the differences in performance amongst the different scheduling rules.

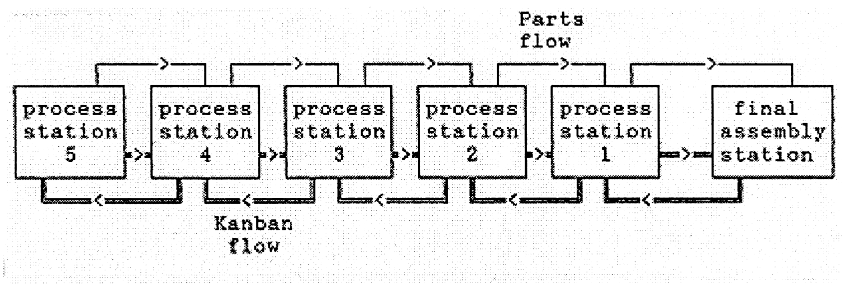

Fig. 1. Flow of kanbans and parts in the system

The model we developed consists of one final assembly station and five processing stations. The final assembly station operates on a predetermined assembly schedule while the five processing workstations operate on the pull principle of the kanban system. The assembly operation is scheduled to alternate among the various final products to approximate the actual working of a mixed-model assembly line.

Processing, on the other hand, will be executed according to the schedule generated by the particular scheduling rule adopted. The processing of parts by a workstation will continue as long as there are kanbans in the queue at the workstation; otherwise the station will stop work and enter into an "idle" state.

A schematic representation of the flow of both kanbans and parts is shown in Fig. 1.

The basic model is structured such that a daily schedule of production would include a specified number of final assemblies of each of the five product types. This will represent a daily workload of 6 h with the chosen assembly times. Most JIT systems are not scheduled to operate for the full 8 h a day as some allowances are normally made for organizational learning, preventive maintenance etc. For our model, the assembly line will stop after the scheduled number of assemblies are completed and when all queues present at the intermediate stations are processed.

The model will begin at an initial state with a specific number of full containers of parts for assembling each of the five types of final products. This ensured that the system will not immediately run into a shortage for parts. This is done as we are seeking to evaluate the performance of the scheduling rules in an ongoing system rather than a startup system. Once these containers of parts are consumed by the assembly stage, the kanbans will move to the upstream station and trigger off production requirements at the preceding workstations.

The structure of the model is also such that each final product and/or intermediate part requires only one container of parts from its immediately preceding station. This avoids unnecessary complexity in terms of parts proliferation. It also allows for control over the number of parts to be scheduled for production at each workstation through the final product-mix assembly schedule.

We also structure our model to provide for a balance of workload across stations. This means that processing times at each of the five process stations for each part type i triggered off by the final assembly of the corresponding final product type i are identical. For example, the final assembly work on one unit of product type A will need 240 s. This will trigger off the need to produce one container worth of part type A at process station 1, which in turn will trigger the need to process part type A at process station 2, etc. And at each of these upstream stations, the processing time needed for part type A is kept identical to the final assembly time of 240 s (that is, a balance of workload over stations).

Further, the model considers the daily demand structure for the final products to be frozen (and hence deterministic) and smoothened out over the production month, as in Toyota's practice of its smoothened master production schedule. This allows for the evaluation of the performance of scheduling rules vis-a-vis each other under stable final demand pull conditions.

Finally, the model in this study assumed zero transportation time for the parts and kanbans between the workstations. This was done to reduce unnecessary complications in the model. We have also excluded features such as lunch breaks and machine breakdowns in the simulation model.

3.3. Performance measures

The scheduling rules will be evaluated on several criteria. We adopted an overall measure as represented by the total completion time, which is the total amount of time required by the system to complete the daily assembly schedule and all outstanding jobs queuing at the processing workstations. This would represent a day's work and preparing the system for the next day's operations. The other performance measures we used included: the average process utilization for the five processing stations, the average number of stoppages per station, and the average duration per stoppage.

3.4. System parameters

Our objective is to evaluate the performance of all four scheduling rules under each given JIT production scenario. We differentiate among the scenarios by the stochasticity of processing times, the extent of setup time reduction, the amount of slack in the system, and the complexity of production requirements in terms of product-mix. These elements therefore serve as the system parameters that we will vary within our simulation model.

3.4.1. Processing times. JIT systems in different stages of implementation will have different degrees of uncertainty in terms of the processing times. We therefore represent this by testing the scheduling rules under these processing time distributions:

(1) Constant processing time (the base case)
(2) Exponential processing time
(3) Narrow normal ($\sigma = 0.1\ \mu$) processing time
(4) Wide normal ($\sigma = 0.5\ \mu$) processing time

The case of constant processing times will be used as the base case for comparison.

3.4.2. Setup times. The second type of JIT scenario represents the extent to which the JIT system has successfully carried out setup time reduction activities. We model this by using the ratio of setup to processing times as follows:

(1) Low setup times (one-tenth of processing times)
(2) Moderate setup times (one-fifth of processing times)
(3) High setup times (one-half of processing times)

3.4.3. System slack (number of kanbans). We represent the amount of slack in the system by the number of kanbans introduced into the system initially. Obviously, with a larger number of kanbans, the work-in-process inventories will be larger and hence the risk of parts shortages will be less, leading to less stoppages at the workstations.

3.4.4. Product-mix. The fourth type of JIT scenario is concerned with the complexity of production requirements placed on the system. We represent this by varying the product-mix and specifying this within the assembly schedule for the final station of our simulation model.

3.5. Experimental design

The following notation is used to represent the system parameters in the various experiments designed for our simulation model (for $i = 1$ to 5):

(1) Processing time of type i parts: P_i
(2) Setup time for type i parts: S_i
(3) Number of kanbans for type i parts: K_i

3.5.1. The base case. The base case scenario simulates a system in a tight production environment. This approximates an ongoing system as opposed to a startup system where there is normally more slack available. The processing times are deterministic and have been set to equal that of the assembly times (i.e a balanced workload across stations).

The base model will have the following parameter values:

(1)	$P_1 = 240$ s	$S_1 = 12$ s	$K_1 = 3$
(2)	$P_2 = 180$ s	$S_2 = 10$ s	$K_2 = 2$

(3)	$P_3 = 120$ s	$S_3 = 12$ s	$K_3 = 3$
(4)	$P_4 = 90$ s	$S_4 = 10$ s	$K_4 = 2$
(5)	$P_5 = 480$ s	$S_5 = 12$ s	$K_5 = 3$

These values are selected and confirmed to be appropriate after experimenting with the test model described at the end of this section. They will also provide a basis for subsequent experiments where all other parameters, except that which is being tested, will be held constant.

3.5.2. Processing time experiments. For the set of experiments on processing times, the mean values for the processing times are held similar to the base case. The coefficient of variation (CV) is 0.1 in the case for narrow normal and 0.5 for wide normal. All other system parameters are as in the base case above. This set of tests will investigate the impact of stochastic processing times on the effectiveness of the scheduling rules.

Ten simulation runs will be made for each experiment involving stochastic processing times. Given that there are four scheduling rules to be evaluated for each scenario, forty simulation runs will have to be performed for each specific situation involving a stochastic processing time. The final analysis and results will be reported in terms of the average obtained from the ten replicated runs. The close replication design of each set of experiments will help to reduce the variances in the results obtained. We adopted this same approach (in terms of number of replications/runs and variance reduction technique) for the other experiments on setup times, number of Kanbans and product-mix which are described next.

3.5.3. Setup time experiments. For this set of experiments, the ratio of setup to processing times will be varied to analyze the effects of varying extent of setup time reduction. The ratios that will be tested are: 1:10, 1:5 and 1:2. The number of setups is expected to become a critical issue when the ratio of setup to processing times increased significantly as it will become more costly to switch production from one part type to another. Since the different rules will schedule jobs differently, we will be able to study the impact of setups on these scheduling rules.

3.5.4. Experiments on the number of kanbans. On this issue, we will gradually increase the number of kanbans from the base case value of three to six. We note that when the number of kanbans is increased for part type one, the same increase is made for the other part types. By increasing the number of kanbans one at a time and keeping all other parameter values constant, the impact of the number of kanbans in the system (and hence the slack in the system) can be analyzed.

3.5.5. Product mix experiments. In this series of tests, we will vary the types of final products from two to five. All other features of the model will remain the same as in the base case. The number of assemblies for each type will be increased as the number of types decreased so that the total number of assemblies (i.e. daily workload) to be produced will be kept constant. By keeping total assemblies (workload) constant, we can attribute any change in performance to product mix and not to an increased workload.

3.5.6. Test case. Prior to all these experiments, we developed a test case under a set of simplified conditions. This involves the case of only two final products where the corresponding deterministic processing times at each of the process stations are much less than the final assembly time for the product type concerned. Under such slack production conditions, we expect there to be no queues and hence the rules should yield the same sequence of production at the processing stations. This test model therefore serves to help verify that the model structure is as desired and the scheduling rules operate in the correct manner. Also, the test model will provide us with a feel for the parameter values to be used for all the experiments explained above.

4. SIMULATION RESULTS AND DISCUSSION

We coded our simulation model using a specialized simulation language: GPSS/H. All the test and experimental runs were executed on an IBM-compatible 386 machine. All results will be reported in terms of the mean values obtained from the many replicated runs. Where stochasticity is present, simple statistical tests of significance will also be reported.

Table 1. Comparison of scheduling rules for base case

	Total time required	Average process utilisation	Average number of stoppages	Average duration of stops
FCFS	30774	83.32	19.8	290.88
Number of Kanbans	25923	88.73	13.2	85.40
Ratio of Kanbans	25649	88.54	13.8	51.57
SPT	24354	89.82	8.6	153.51

4.1. Base case

In the base case, we have five products scheduled in the final assembly schedule. The processing time for each type of parts is deterministic and equal to the assembly time for the corresponding final assembly (this simulates a balance of workload across stations). The setup time is relatively low (simulating a JIT environment with setup time reduction well underway) and the number of kanbans is relatively few (representing a fairly tight production environment). These conditions were selected to approximate the production conditions of an ongoing system rather than that of a startup production system.

The results from the base case are as presented in Table 1. The performance measure reported on the total time is in seconds and it represents the time taken for the final assembly schedule to be completed as well as all outstanding queues of kanban at each station. The average process utilization across the five processing stations is expressed as a percentage. The average number and duration of stoppages are measured across all six stations.

In this case, FCFS performed very poorly as compared to the other three rules. The large number of stoppages for FCFS was largely due to its inability to consider the risk of parts shortage. This resulted in frequent stoppages of considerable duration. Both the Kanban related rules performed quite well in terms of total required time and process utilization, but the Ratio of Kanban rule seems better with a shorter duration of overall stoppage time.

On the other hand, the SPT scheduling rule performed best in terms of the total completion time. SPT performed well in this case because of the manner with which assembly was carried out. The frequent switching of production from one type of final product to another meant that there was no prolonged demand for any one particular type of parts. In such a scenario, the SPT rule was able to schedule in a manner to hasten the pace of parts clearance from the processing stations. Also, it should be noted that this result is similar to that in Lee [1] where the SPT/LATE rule was highlighted as the best overall rule. It is also consistent with that of Yavuz and Satir [25] where the SPT was stated as the best general rule.

In summary, for an ongoing JIT system, with a balanced workload across stations, well reduced setup times, a minimum number of Kanbans representing a minimum of slackness in terms of the work-in-process inventory, and deterministic processing times, our simulation results suggest the following:

i) The choice of scheduling rule is important. The difference between the commonly practised rule of FCFS and the best performing rule of SPT here is approximately 6000 s (i.e. 100 min).

ii) The SPT rule performs well in an ongoing JIT mixed-model assembly environment with no stochasticity (minimum variability of processing times).

iii) However, the difference in performance between the Kanban related rules and that of SPT is much less (about 1500 s or 25 min).

The implication of the above results is that for an ongoing JIT system, the commonly adopted FCFS rule does not appear to be a good choice. The SPT rule may be a better one.

4.2. Processing time experiments

In this set of experiments, the processing times were allowed to take on different statistical distributions. All other parameters remain as in the base case above.

Table 2. Comparison of scheduling rules under exponentially distributed processing times

	Mean total completion time	Standard deviation	Average process utilization	Average number of stops	Average duration of stops
FCFS	38632.2	4037.31	70.72	14.9	482.87
Number of Kanban	32997.0	3081.48	73.38	13.8	303.06
Ratio of Kanban	33881.6	3826.27	73.44	13.5	279.57
SPT	34856.0	3738.02	74.20	13.8	382.30

4.2.1. Exponential distribution. When the processing times were exponentially distributed, the Number of Kanbans rule performed better than the other three rules while the frequently adopted FCFS rule again performed worst with regards to mean total completion time. The difference was substantial and amounted to an average of close to 6000 s (see Table 2). This difference between the best and worst rule was tested and found to be statistically significant at the 5% level using the Student's t-test for matched pairs.

Both the Number of Kanban and the Ratio of Kanban rules were able to perform well due to their consideration of demand for the parts. FCFS, which merely scheduled production according to the arrival pattern, caused the system to encounter more stoppages and hence took more time to complete the daily workload.

The SPT rule, which had performed well in the base case, no longer was the best. This was due to the large variation in the processing times under the exponential distribution. The faster jobs that were given priority might in fact require much longer than the mean job time. This would lead to long queues being created at the processing stations, eventually causing stoppages in the system.

4.2.2. Narrow normal distribution. As in the case of exponential processing times, the FCFS rule performed poorly as compared to the other three scheduling rules. In this case, the Ratio of Kanban rule had the best results with the shortest mean completion time (see Table 3). The difference between the completion times for FCFS and Ratio of Kanban rules was again found to be statistically significant at the 5% level.

The ability of the Ratio of Kanban rule to consider the likelihood of stoppage was clearly evident in this case and helped to greatly reduce total completion time. The stoppages, when they did occur, were of a significantly shorter duration when compared to the other three scheduling rules.

The SPT rule also performed well in this scenario due to the relatively small fluctuations in processing times. With this small variation in processing times, the preference for scheduling parts with shorter processing times would not result in many stoppages as parts that were given higher priority would indeed be completed relatively quickly and queues would not grow at the processing stations.

4.2.3. Wide normal distribution. In the case of the simulation experiments for the wide normal distribution, the coefficients of variation of processing times were increased from 0.1 to 0.5. The results obtained indicated that the differences in the performance of the four scheduling rules were greater than that

Table 3. Comparison of scheduling rules under narrow normal distribution

	Mean total completion time	Standard deviation	Average process utilization	Average number of stops	Average duration of stops
FCFS	31305.1	309.01	76.94	16.7	201.22
Number of Kanbans	26035.1	484.59	85.11	12.0	103.77
Ratio of Kanban	25688.8	429.34	88.43	11.9	60.58
SPT	26027.9	512.91	86.98	8.5	125.68

Table 4. Comparison of scheduling rules under wide normal distribution

	Mean total completion time	Standard deviation	Average process utilization	Average number of stops	Average duration of stops
FCFS	33426.8	840.81	76.18	15.9	266.55
Number of Kanbans	29273.7	1160.66	81.04	13.7	161.03
Ratio of Kanban	28699.1	1087.68	84.11	13.0	124.60
SPT	29535.0	1606.69	82.41	11.2	223.32

obtained for the narrow normal case (see Table 4).

As before, FCFS performed worst among the four different scheduling rules and the Ratio of Kanban rule once again had the best results. The superiority of the Ratio of Kanban rule over the FCFS rule in terms of total completion time was again confirmed to be statistically significant at the 5% level.

The SPT rule fared relatively poorly in this case due to the wide fluctuation in processing times. In contrast to the case of narrow normal distribution, where the SPT rule was only slightly inferior to the Ratio of Kanban rule, the SPT rule in this scenario was significantly poorer (at the 5% level) than the Ratio of Kanban rule in terms of mean total completion time. It should be noted that the confidence level for these two statistical statements to be true simultaneously is at least 90% (Bonferroni's Inequality).

For the wide normal case, the SPT rule which favors the scheduling of parts with shorter processing times might cause queues to develop as the parts selected might in effect take much longer than expected. Hence, it can be seen that the SPT rule would perform poorly in situations where there are large fluctuations in processing times as indicated by its poor showing in both the exponential and wide normal cases.

In summary, our results seem to indicate the following:

i) Under deterministic processing times, the SPT rule performs well (as seen in the base case).

ii) Where there in stochasticity in processing times, the Ratio of Kanban rule tends to outperform the SPT rule and is the most effective rule amongst the four rules.

iii) The greater the stochasticity in processing times (higher coefficient of variation), the better the performance of the Ratio of Kanban rule relative to the other rules.

iv) The commonly adopted FCFS rule is consistently the worst performing rule according to our results.

It should also be highlighted that our results here are consistent with that obtained by Lee and Seah [2] where they found that the FCFS and SPT/LATE rules performed better under conditions of constant and normal processing times (low coefficient of variation) as compared to negative exponential processing times (higher coefficient of variation). Lee and Seah [2] also highlighted the FCFS rule as their worst rule in terms of overall system performance.

The implication of the above results is that with a recently started JIT line where processing times are still highly variable, the Ratio

Table 5. Comparison of scheduling rules when setup equals one-tenth of processing time

	Total time required	Average process utilization	Average number of stoppages	Average duration of stoppages
FCFS	31732.0	83.96	19.8	323.10
Number of Kanbans	26962.0	87.62	13.8	102.01
Ratio of Kanban	26986.0	87.78	13.6	82.65
SPT	24801.0	90.30	8.8	191.66

Table 6. Comparison of scheduling rules when setup equals one-fifth of processing time

	Total time required	Average process utilization	Average number of stoppages	Average duration of stoppages
FCFS	33439.0	85.04	19.8	373.97
Number of Kanbans	28722.0	88.36	14.8	114.41
Ratio of Kanban	28783.0	88.02	16.0	94.59
SPT	26161.0	90.54	9.6	197.17

of Kanban rule should be adopted. However, as the system begins to stabilize in terms of reduction in variability of processing times, the SPT rule may be preferred. The commonly used FCFS rule does not appear to be a wise choice.

4.3. Setup time experiments

For this set of experiments, we varied the ratio of setup to processing times to analyze the impact of setup time on scheduling rules effectiveness. Apart from this, all other parameters were as in the base case. When setup time was one-tenth of processing time, the following results were obtained (see Table 5).

The FCFS rule required the most time in order to complete all the work while the SPT rule had the best results. FCFS resulted in frequent changes in the type of parts to be produced and hence more setups. The SPT rule which gave a higher priority to parts with a shorter processing time would reduce the total number of setups as there would be less frequent changeovers. A similar reasoning provides the explanation for the weaker performance of the Kanban related rules relative to the SPT rule.

When the ratio of setup to processing times was increased from 1:10 to 1:5, the following results were obtained (see Table 6).

SPT again achieved higher process utilization, fewer stoppages and lower total completion times than all the other three rules. The FCFS rule again proved to be the worst of the four different scheduling rules. This was similar to the earlier case of lower setup times: there was no change in the ranking of the four scheduling rules in terms of total completion time required.

For the next set of simulation experiments, we proceeded to further increase the setup times to one-half of processing times. The following results were obtained (see Table 7).

Again, similar results were obtained with SPT performing well while FCFS faring poorly with respect to the other rules. One point to note is the increasing difference in the total completion time between the best and worst rules as the ratio of setup to processing times increased (from a difference of about 6000 s in Table 5 to 7600 s in Table 7).

In summary, our results in this set of experiments based on differences in setup times seem to suggest the following:

i) As setup to processing time ratio increases, the superiority of the SPT rule becomes obvious relative to the other rules, and especially relative to the commonly adopted FCFS rule.

ii) Since deterministic processing times were used here (base case condition), it is also clear that SPT performs best under processing times with zero variability.

The implication from this set of experiments on setup times is that for an ongoing JIT system, when setup times are high (before vast improvements in setup time reductions), SPT is clearly better than the commonly adopted

Table 7. Comparison of scheduling rules when setup equals one-half of processing time

	Total time required	Average process utilization	Average number of stoppages	Average duration of stoppages
FCFS	39721.0	85.06	20.0	531.52
Number of Kanbans	35250.0	89.28	17.4	188.10
Ratio of Kanban	34335.0	90.28	18.0	140.67
SPT	32100.0	87.74	5.4	211.63

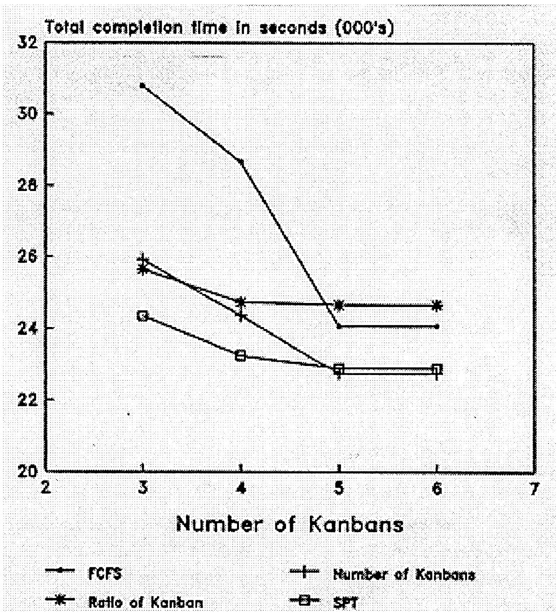

Fig. 2. Impact of number of kanbans on total completion time

FCFS rule. Quite clearly, our results here add to the existing literature since only Lee and Seah [2] have previously considered the effects of setup times. Even then, they only evaluated the performance of their SPT/LATE rule under varying setup times. They did not compare the relative performances of the different scheduling rules.

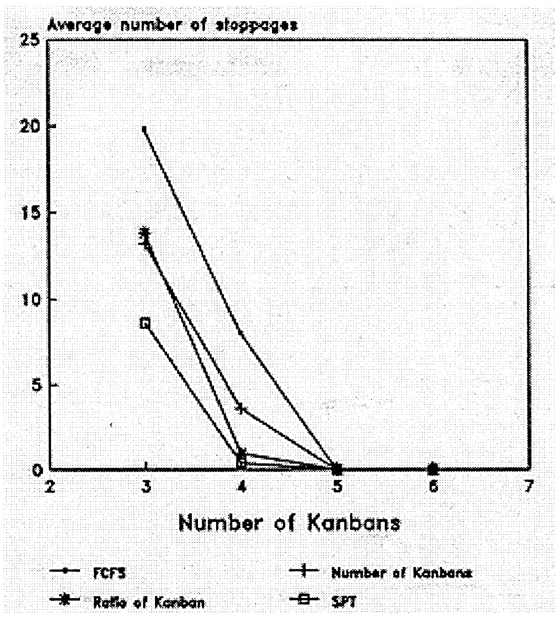

Fig. 3. Impact of number of kanban on the number of stoppages

4.4. Number of kanbans experiments

The results obtained for changes in total completion time with changes in the number of kanbans are as shown in Fig. 2. Initially, when the number of kanbans was relatively low (representing minimum slack environment), the introduction of an additional kanban for each type of parts reduced total completion times. As the number of kanbans gradually increased, the system became less sensitive to further increases in the number of kanbans. This was most apparent in the case when the number of kanbans was increased from five to six where the results indicated no change in the total completion times.

From the graph, it can also be noted that the reduction in total completion time was most significant for the FCFS rule where the decrease in total completion time was in excess of 6000 s when the number of kanbans was increased from three to five. On the other hand, the other three scheduling rules were not as sensitive to the number of kanbans, with reductions of between 2000 to 3000 s. This indicated that these rules were more effective when the system was tight and the increase in the kanban had relatively less impact than in the case for the FCFS rule.

Also, it may be noted that the difference in performance amongst the four rules narrowed as the number of kanbans increased. Conversely, as the amount of slack is decreased, the difference in performance widened.

Besides the change in total completion time, there was also a fall in the number of stoppages as the number of kanbans in the system increased. This is indicated in Fig. 3. All the four different scheduling rules had sharp falls in the number of stoppages when the number of kanbans was increased from three to four. As soon as the number of kanbans was five, all the four rules had no stoppages and the further addition of one more Kanban had no impact in terms of stoppages. The Number of Kanban rule and Ratio of Kanban rule had initially quite similar number of stoppages but the difference widened as soon as one more kanban was added. The Ratio of Kanban rule was more sensitive and enjoyed a larger reduction in the number of stoppages than the Number of Kanban rule.

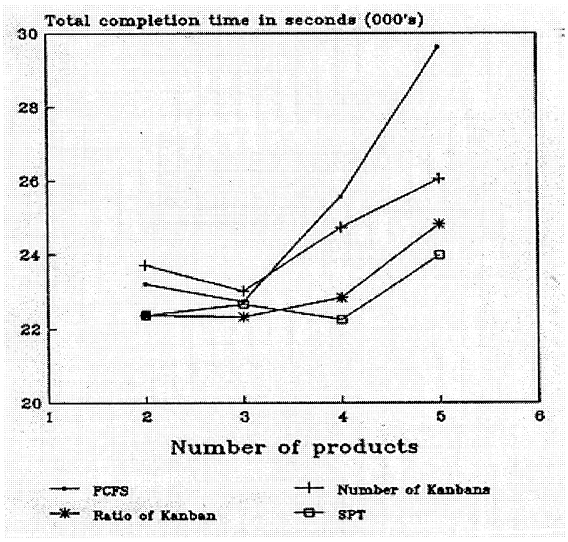

Fig. 4. Impact of product mix on total completion time

Again, performance differences amongst the four rules in terms of the number of stoppages narrowed as the number of kanbans increased, till when considerable slack existed and stoppages ceased altogether.

In summary, from this set of experiments, we can observe the following:

i) The FCFS rule is most sensitive to the number of kanbans in the system whereas the SPT rule is least sensitive to this parameter; generally, the SPT rule seemed to yield the best peformance.

ii) In terms of total completion time, the differences amongst the four rules reduced as the number of kanbans increased.

iii) In terms of the number of stoppages, the differences also narrowed as the number of kanbans increased.

The implication from the results obtained above is that the number of kanbans (i.e. slackness in the system) affects the relative performance of the scheduling rules and the choice of a scheduling rule is more significant when the system is tighter, which is highlighted by the larger differences amongst the four rules when the number of kanbans is lower. Our results here are therefore consistent with that of Yavuz and Satir [25] where they reported the generally superior performance of the SPT rule especially in more restrictive operational environments (that is, when the system is tighter). Also, Berkley [23] showed that the SPT rule performed better than the FCFS rule when buffer capacities are smaller (fewer number of kanbans).

4.5. Product mix experiments

Here we studied the impact of product mix by varying the number of final assemblies from two to five types while other parameters remained as in the base case. The results obtained on total completion time are presented in Fig. 4.

The increase in total completion time was highest for the FCFS rule while the impact on the other three rules was much less. This was expected as the FCFS rule scheduled production according to arrivals and hence would in most instances be scheduling production of parts that were not immediately required by the downstream station. This would result in the delay in production of parts that were more critically needed, thereby leading to stoppages in the system. There was little change in the total completion time for the SPT rule as the product mix increased from two to five. This was also true for the Ratio of Kanban rule. This indicated that the latter two scheduling rules were less sensitive to the product mix.

The differences in completion time when the number of products was two and three were much less than when the number of products increased to four and five. While the Ratio of Kanban and SPT rules continued to perform well when the number of products increased, the Number of Kanban and FCFS rules deteriorated in their performance.

Generally, the number and duration of stoppages increased for all four rules as the product mix increased. The rule that performed best in terms of all the performance measures was the SPT rule. The SPT rule was more able to schedule production to meet demand at the downstream station.

An interesting finding of Fig. 4 is the initial decrease in total completion time for the FCFS and NKB rules (when changing from 2 to 3 products) and for the SPT rule (when changing from 3 to 4 products). A possible explanation for this is that the initial increase in the number of final products means a longer time interval before any given product/part is pulled again so that stoppages and hence the completion time may be reduced. However, as the number of products increased further, the scheduling choices and burden begin to dominate so that stoppages increase and hence the total completion times.

In summary, the results from our experiments on product mix suggest the following:

i) As product mix increases, the performance of all the four scheduling rules deteriorates.

ii) There is a difference in the sensitivity to product mix changes among the four scheduling rules. FCFS rule is most sensitive to the number of final products whereas the SPT rule is least sensitive.

iii) The difference in relative performance widens as the product mix increases.

The implication from the above is that the choice of scheduling rule is more important in a wide product mix environment, with the SPT rule being a good choice in such a setting. This result is also consistent with that of Yavuz and Satir [25].

5. CONCLUSION

Our simulation results reported above seem to suggest the following. Firstly, users of JIT systems should not arbitrarily adopt a scheduling rule. Instead, they should seek to understand the nature of different scheduling rules and the production environment. Secondly, the use of the FCFS rule does not appear to be justified. The weakness of the FCFS rule becomes even more apparent under tight production conditions. Thirdly, users need to match the scheduling rule with the conditions that they face. Different production scenarios will require different scheduling rules to achieve best overall system performance. More specifically, our results seem to indicate that the choice of rules should be made with reference to the nature of the processing times. In situations where the processing times are deterministic (or have low variances), the SPT rule outperforms the other three scheduling rules. In scenarios where processing times are stochastic with significant coefficients of variation, the Ratio of Kanban (RKB) rule performs best in reducing total completion time and minimizing stoppages. It also appears that the superiority of the RKB rule increases as the stochasticity in processing times increases.

Finally, and most importantly, our results implied that the need to choose a scheduling rule is greater when the production scenario is characterized by little or no slack in the system, a relatively large product-mix, and setup times which have not been significantly reduced relative to processing times. This is because the differences in the performance of the four rules appear to increase with the tightness of the system, the product-mix, and also the ratio of setup to processing times.

Our results are based on the study of a six-station kanban-controlled serial line, evaluating a total of four scheduling rules operating under different production conditions. We have seek to add to both the scope and depth of earlier work in this area of JIT scheduling: thus, while we have added only one new scheduling rule (the RKB rule), we have, however, subjected all the four rules to a common, comprehensive set of different operating environments for comparative evaluation; and while our results are apparently consistent with existing results for each separate experimental factor, the focus of our results has been on the relative performance of all four rules; and more specifically, our study has also yielded significant results on the performance of the new RKB rule which seemed especially relevant in the context of processing time stochasticities. Further related work could extend such studies to other structures of JIT systems (as opposed to the serial line structures studied so far). Similarly, other product structures could be studied and other JIT-unique scheduling rules could be developed and evaluated.

REFERENCES

1. Lee, L. C., Parametric appraisal of the JIT system. *International Journal of Production Research*, 1987, **25,** 1415–1429.
2. Lee, L. C. and Seah, K. H. W., JIT and the effects of varying process and set-up times. *International Journal of Operations and Production Management*, 1988, **8,** 19–35.
3. Berkley, B. J. and Kiran, A. S., A simulation study of sequencing rules in a kanban-controlled flow shop. *Decision Sciences*, 1991, **22,** 559–582.
4. Berkley, B. J., Simulation tests of FCFS and SPT sequencing in kanban systems. *Decision Sciences*, 1993a, **24**, 218–227.

5. Sugimori, Y., Kusunoki, K., Cho, F. and Uchikawa, S., Toyota production system and kanban system materialization of JIT and Respect-For-Human system. *International Journal of Production Research*, 1977, **15,** 553–564.
6. Schonberger, R. J., *Japanese Manufacturing Techniques in Nine Hidden Lessons in Simplicity*. The Free Press, New York, 1982.
7. Monden, Y., *Toyota Production System: a Practical Approach to Production Management*. Industrial Engineering and Management Press, Atlanta, 1983.
8. Cheng, T. C., The JIT production: a survey of its development and perception in the Hong Kong electronics industry. *Omega*, 1988, **16,** 25–32.
9. Crawford, K. M., Blackstone, J. H. and Cox, J. F., A study of JIT implementation and operating problems. *International Journal of Production Research*, 1988, **26,** 1561–1568.
10. Hum, S. H. and Ng, Y. T., A study on JIT practices in Singapore. *International Journal of Operations and Production Management*, 1995, **15,** 5–24.
11. Im, J. H. and Lee, S. M., Implementation of JIT systems in US manufacturing firms. *International Journal of Operations and Production Management*, 1989, **9,** 5–14.
12. Voss, C. and Robinson, S., Application of JIT manufacturing techniques in UK. *International Journal of Operations and Production Management*, 1987, **7,** 46–52.
13. Huang, P. Y., Rees, L. P. and Taylor, B. W., A simulation analysis of the Japanese JIT technique (with kanbans) for a multiline. *Multistage Production System, Decision Sciences*, 1983, **14,** 326–343.
14. Jordan, S., Analysis and approximation of a JIT production line. *Decision Sciences*, 1988, **19,** 672–681.
15. Kim, T. M., JIT manufacturing system: a periodic pull system. *International Journal of Production Research*, 1985, **23,** 553–562.
16. Miltenburg, J. and Sinnamon, G., Scheduling mixed-model multi-level JIT production systems. *International Journal of Production Research*, 1989, **27,** 1487–1509.
17. Philipoom, P. R., Rees, L. P., Taylor, B. W. and Huang, P. Y., A mathematical programming approach for determining workcentre lotsize in a JIT system with signal kanbans. *International Journal of Production Research*, 1990, **28,** 1–15.
18. Keller, A. and Fouad, R., A review of literature relating to JIT. *International Journal of Operations and Production Management*, 1989, **9,** 15–25.
19. Golhar, D. and Stamm, C. L., The Just-in-time philosophy: a literature review. *International Journal of Production Research*, 1991, **29,** 657–676.
20. Korgaonker, M., *Just-in-time Manufacturing*. MacMillan India Press, Madras, 1992.
21. Lummus, R. R., A simulation analysis of sequencing alternatives for JIT lines using kanbans. *Journal of Operations Management*, 1995, **13,** 183–191.
22. Philipoom, P. R., Rees, L. P. and Taylor, B. W., Simultaneously determining the number of kanbans, container size, and the final-assembly sequences of products in a Just-in-time shop. *International Journal of Production Research*, 1996, **34,** 51–69.
23. Berkley, B. J., Effect of buffer capacity and sequencing rules on single-card kanban system performance. *International Journal of Production Research*, 1993b, **31**, 2875–2893.
24. Chao, X., Das, S. K. and Nagendra, P., Prioritization of kanbans in the case of a single station serving multiple downstream stations. *International Journal of Production Research*, 1995, **33,** 377–389.
25. Yavuz, I. H. and Satir, A., A kanban-based simulation study of a mixed model Just-in-time manufacturing line. *International Journal of Production Research*, 1995, **33,** 1027–1048.

ADDRESS FOR CORRESPONDENCE: *Prof. Sin-Hoon Hum, Department of Decision Sciences, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Republic of Singapore.*