Analyzing Part Scheduling and Tool Usage in a Flexible Manufacturing System Using Simulation

Karuna Kumar .G[1], .[2]Dayakar - 22485A0331,.[3]Vijaya raj - 22485A0330,[4]D.Riyazuddin - 21481A0377 ,[5]K.Bharath kumar - 21481A0354,[6]Dr. R. Sridharan
[1]Assistant professor, Department of Mechanical Engineering, Seshadri Rao Gudlavalleru Engineering College , Gudlavalleru, Andhra Pradesh 521356India.
[2,3,4,5] students, Department of Mechanical Engineering, Seshadri Rao Gudlavalleru Engineering College ,Gudlavalleru, Andhra Pradesh 521356India.
[6]Professor Dept. of Mechanical Engineering NIT Calicut NIT Campus P.O. Calicut-673601, Kerala, India.

---***---

Abstract

In this study, simulation is employed to examine the impact of part scheduling and tool sharing decisions within a Flexible Manufacturing System (FMS) that incorporates both part and tool movement policies. A representative FMS configuration is modeled, comprising a load/unload station, three machine groups (each containing two identical machines), dedicated tool stores for each group, three Automated Guided Vehicles (AGVs) for part transport, and individual tool transporters assigned to each machine group. The system is designed to handle 15 distinct part types, each characterized by a unique processing sequence and operation-specific processing times. Part arrivals follow an exponential inter-arrival time distribution with a predefined mean. Simulation experiments are carried out across various combinations of scheduling and tool management decision rules. Key performance indicators evaluated include Mean Flow Time, Mean Tardiness, Mean Tool Waiting Time, and Mean Machine Utilization. The results are analyzed to identify the most effective decision rule combinations for optimizing system performance.

Keywords: *Flexible job shop, Scheduling, Simulation, Job release.*

1. Introduction

Flexible job shop scheduling problem is an important extension of the classical job shop scheduling problem. The classical job shop scheduling problem involves determining a schedule for jobs that have pre-specified operation sequences in a multi machine environment. In the classical static job shop scheduling problem, *n* jobs have to be processed to completion on *m* machines. For each job, technological constraints specify a complete, distinct routing which is fixed and known in advance. Processing times are sequence independent, fixed, and known in advance. Each machine is continuously available from time zero, and operations are processed without preemption. In order to meet today's market requirements like more product variety and shorter lead time, manufacturing systems have to become more flexible and efficient. In a flexible job shop system, the machines are flexible due to the presence of a considerable amount of overlapping capability to perform a variety of operations. As such, an operation can be performed on more than one machine. This leads to a complex problem known as the flexible job shop scheduling problem. The dynamic version of flexible job shop scheduling problem can be stated as follows: The job shop consists of a set $U = \{M_1, M_2, ..., M_m\}$ of machines. Jobs for processing in the system arrive continuously over time in a random manner. A job *i* consists of a sequence $Oi_1, Oi_2, ...Oi_{ki}$ of operations to be performed one after the other according to the given sequence. Each operation O_{ij} can be executed on any among a subset $U_{ij} \subseteq U$ of compatible machines. There is partial flexibility if there exist a proper subset U_{ij} of U, for at least one operation O_{ij}, while $U_{ij} = U$ for each operation O_{ij} in the case of total flexibility. The processing time of each operation is machine dependent. The scheduling problem involves two sub problems:

Kacem et al. classify the FJSSP into two subproblems as follows:

- Total FJSSP (T-FJSSP): each operation can be processed on any of the machines '*m*' in the shop.
- Partial FJSSP (P-FJSSP): each operation can be processed on one machine of subset of the machines '*m*' in the shop.

According to Kacem et al. for the same number of machines and jobs, the P-FJSSP is more difficult to solve than the T-FJSSP. Therefore, Kacem et al. [1] suggest that the P-FJSSP can be transformed to the T-FJSSP by adding infinite processing times to the unused machines. The Static Flexible Job Shop Scheduling Problem (SFJSSP) has been investigated by several researchers. The flexible job shop scheduling problem is more difficult than the classical job shop scheduling problem since it involves machine selection decision in addition to the job scheduling decision. Exact algorithms for solving SFJSSP are not effective for solving larger size problems. Hence, heuristic methods such as dispatching rules and met heuristics (tabu search, simulated annealing and genetic algorithms) are widely used for SFJSSP. Brucker et al. analyze the computational complexity of FJSSP.

Two types of heuristic approaches are found in the literature for solving SFJSSP: hierarchical approach and integrated approach. In the hierarchical approach, the problem is solved by decomposing it into a sequence of sub-problems. A typical decomposition is assign-sequence method. Brandimarte and

Paulli have adopted this approach wherein the assignment problem is solved using dispatching rules and the resulting sequencing problem is solved using tabu search heuristics. Integrated approach is in general found to provide better results though it is much more difficult. Vaessens et al. and Dauzere-Peres and Paulli adopt an integrated approach using tabu search method. Chen et al. Kacem et al. Jia et al. Ho et al. , Chan et al. and Pezzella et al have used integrated approach employing genetic algorithms. Saidi-Mehrabad and Fattahi [12] present a two-phase algorithm that uses a tabu search procedure for minimizing makespan. The first phase involves searching for the best sequence of operations of jobs while the second phase involves a procedure that finds the best choice of machine alternatives. Ho et al. describe a practical model of flexible job shop with an example based on job assignments encountered in the manufacturing of printed circuit boards.

The objective of the research study reported in this paper is to investigate the effect of job release policies on the performance of a typical dynamic flexible job hop system (DFJSS). The system considered consists of eight individual machines. The machines are capable of performing a variety of operations. The case of partial flexibility is considered wherein an operation can be executed on three different machines. Each operation can be performed efficiently on the primary machine. The other two alternative machines are also capable of performing the same operation though less efficiently. This is modelled as a percentage increase in processing time when an operation is performed on an alternative machine. The job release policies evaluated are immediate release and flow allowance based release. A discrete event simulation model is developed for describing the working of the system. Scheduling rules from the literature are used for the decisions such as machine selection and job scheduling. The performance measures evaluated are mean flow time, standard deviation of flow time, mean tardiness, standard deviation of tardiness, and percentage of tardy jobs. A discrete event simulation model is developed for describing the working of the system. The performance measures evaluated are mean flow time, standard deviation of flow time, mean tardiness, standard deviation of tardiness, and percentage of tardy jobs. The performance measures are determined using the simulation results obtained after the system reaches steady state. The results have been subjected to graphical analysis.

The rest of the paper is organized as follows: Section 2 describes the job release policies. Section 4 describes the flexible job shop system. Section 5 provides the details of the simulation model. Section 6 presents the experimentation

The details of the study are presented in the following sections.

2. Job Release Policies

In the present study, the following policies used for releasing the jobs into the system for processing:

- Immediate Release Policy:

In this policy, when a job arrives to the system, it is released immediately into the system for processing. This means that job release time is equal to job arrival time.

- Flow Allowance based Release Policy:

In this policy, the release time of a job is calculated using the due-date and the flow allowance. Release time R_i of job i is computed as follows:

$R_i = d_i - F$

where, R_i = Release time of job i; F = Flow allowance; d_i = Due date of job i.

In the present study, the flow allowance is estimated from a preliminary study and it is fixed as equal to 250 minutes.

3. System Configuration

The system analyzed in this study includes eight machines, each with distinct capabilities, allowing them to carry out a variety of operations. In total, the system can handle 20 unique operation types. For each operation, there are three machines that can perform it, as detailed in Table 1. These machines are categorized as Primary (1), Secondary (2), and Tertiary (3) based on their suitability for the specific operation.

Table 1 Compatible Machines for Various Operations

Operation Type	M1	M2	M3	M4	M5	M6	M7	M8
OP1		1			2		3	
OP2			3		1	2		
OP3		3		2				1
OP4	2			3		1		
OP5		2	1			3		
OP6	1						2	3
OP7	3	1	2					
OP8				1		3		2
OP9		2			1		3	
OP10			2		3		1	
OP11	2		1			3		
OP12	2				3	1		
OP13	3		2					1
OP14			3	2			1	
OP15	1	2		3				
OP16				1		2		3
OP17			3		2		1	
OP18		3			2	1		
OP19				3			2	1
OP20	3	1						2

It is considered that the processing time of an operation on secondary and tertiary machines is more than that of the primary machine by 10% and 20% respectively.

Jobs arrive at the system according to a Poisson process with the interarrival time following exponential distribution. The mean of the exponential distribution is set equal to 21 minutes in the simulation experiments. The number of operations required for each job is uniformly distributed in the range 5-8.

For each operation, the type of the operation is determined by random assignment among the 20 different types of operations that can be processed in the system. The operation type of an operation of a job is established in such a manner that an operation type is included in the sequence of operations (process plan) not more than once. Processing time of an operation of a job on the primary machine is generated using an exponential distribution with a mean of 30 minutes. Due date of each job is determined using the TWK method, as follows.

$d_i = a_i + K\sum_{j=1}^{ki}\overline{p_{ij}}$, where a_i is arrival time of job i, d_i = due-date of job i, k_i is the set of operations required for job i and $\overline{p_{ij}}$ is the average processing time of operation j of job i.

$\overline{p_{ij}} = \dfrac{p_{ij} + 1.1p_{ij} + 1.2p_{ij}}{3} = 1.1\ p_{ij}$ where p_{ij} is the processing time of operation j of job i on primary machine. The due-date factor K is set equal to 2.

4 Simulation Model

A discrete-event simulation model is developed to describe the operation of the flexible job shop system. The coding of the simulation model involves incorporating the necessary logic. The simulation model consists of modules such as the data generation, event routine, machine assignment, job scheduling and report generation.

4.1 Machine selection Module

The choice of the machine for processing an operation of a job is based on the following scheduling rule.

EFTA - Earliest Finishing Time with Alternatives

Select the machine with the least value of the sum of the following three quantities.

- Sum of the processing time of the operations of jobs waiting in the queue of the machine (Work load of the machine).
- Remaining processing time of the machine for completing the current operation.
- Processing time of the operation of the job that is considered for assignment.

Thus, EFTA denotes the earliest time at which the operation (to be assigned) will be completed on the machine, if the operation is assigned to the machine.

4.2 Job Scheduling Module

This module contains the logic for the scheduling rules used for selecting the job to be processed on a machine from among the jobs waiting in the queue of the machine. In the present study, four existing rules from the literature are used. These scheduling rules are explained as follows.

- Shortest Processing Time (SPT)

Select the job with the shortest processing time for the imminent operation.

- Shortest Remaining Processing Time (SRPT)

Select the job with the shortest remaining processing time of the job. The remaining processing time for a job is the sum of the average processing time of all the remaining operations.

- EDD: Earliest Due-Date

Select the job with the smallest due-date.

- EMDD: Earliest Modified Due-Date

The modified due date for a job is the job's original due-date or its early finish time whichever is larger. Modified Due-Date (MDD) = Max {due-date, (current time + remaining processing time)}

4.3 Performance Measures

The report generation module provides the values of the performance measures such as mean flow time, standard deviation of flow time, mean tardiness, standard deviation of tardiness, and percentage of tardy jobs.

5 Simulation Experiments

The first stage in the simulation experimentation is determining the end of the initial transient period (identification of the steady state). For this purpose, Welch's procedure described in Law and Kelton [15] is used. It is a graphical procedure consisting of plotting moving averages for the output performance measures. The end of the initial transient period is the time at which the moving averages approach a level value. In the present study, a pilot simulation study has been conducted for this purpose. Ten replications are made. Each replication consists of simulating the operation of the system for the completion of 1000 jobs. The experimental setting used is as follows. Mean interarrival time of jobs: 21 minutes; Job scheduling rule: SPT. It is found that the moving averages for all the performance measures approach a level value when 250 jobs are completed. In the simulation experiments, the simulation for each replication is run for 1250 job completions. Jobs are numbered on arrival at the system and the simulation outputs from jobs numbering 1 to 250 (transient period) are discarded. The outputs for the remaining 1000 jobs (jobs numbering 251 to 1250) are used for the computation of the performance measures.

In the present study, four scheduling rules are used for the scheduling jobs for each of the two job release policies. Thus, there are eight simulation experiments. Ten replications are made for each simulation experiment. The values of the performance measures such as mean flow time, standard deviation of flow time, mean tardiness, standard deviation of tardiness and percentage of tardy jobs are determined using the simulation output after the system reaches steady state. For each experiment, the simulation output for the ten replications is averaged. These values are presented in Figures 1 to 5.

6. Results and Discussion

The simulation results are analyzed using the graphical plots.

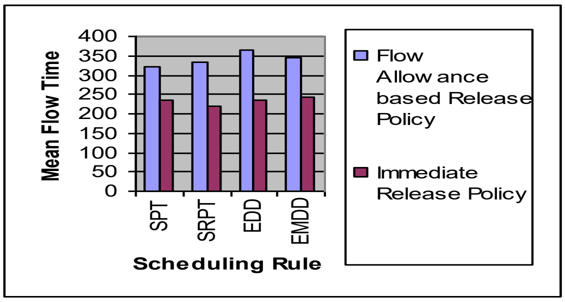

Figure 1: Mean Flow Time

Figure 1 shows the variation of mean flow time for the two job release policies. The scheduling rule SRPT is found to provide better results for immediate release policy while SPT provides better values for flow allowance based release policy. Further, all the scheduling rules provide almost similar values for the immediate release policy.

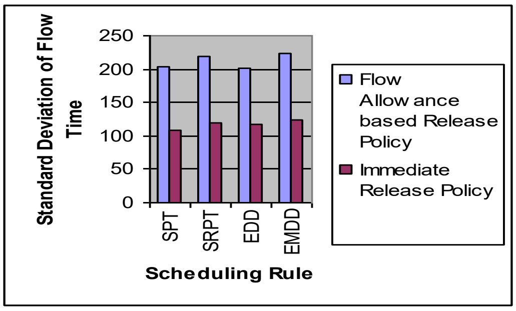

Figure 2: Standard Deviation of Flow Time

Figure 2 depicts the variation of standard deviation of flow time. The scheduling rule SPT is found to provide better results for immediate release policy while EDD provides better values for flow allowance based release policy. As observed for the mean flow time measure, all the scheduling rules provide almost similar values for standard deviation of flow time for the immediate release policy.

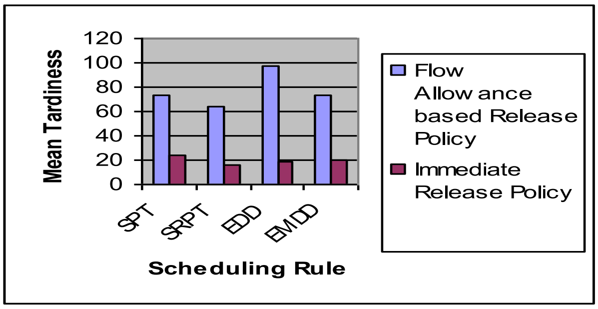

Figure 3: Mean Tardiness

The simulation results for the mean tardiness measure are shown in Figure 3. The scheduling rule SRPT is found to provide smaller values for both the release policies. It is also observed that immediate release policy provides smaller values when compared with the flow allowance based release policy.

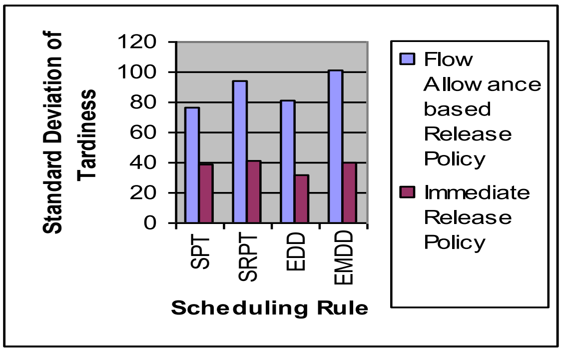

Figure 4: Standard Deviation of Tardiness

Figure 4 shows the variation of standard deviation of tardiness. For immediate release policy, EDD rule is found to provide better results; all the other scheduling rules provide almost same value for this policy. SPT provides better values for flow allowance based release policy. Also, there is a considerable variation in standard deviation of tardiness among the scheduling rules when flow allowance based release policy is adopted.

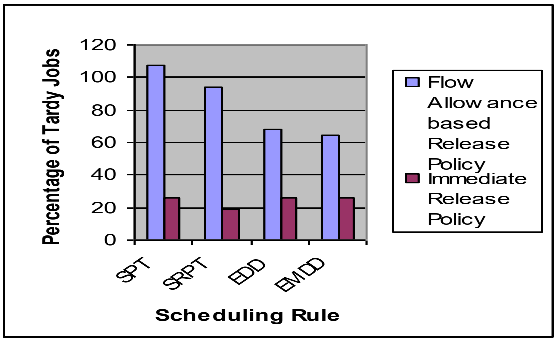

Figure 5: Percentage of Tardy Jobs

Figure 5 provides the percentage of tardy parts for the simulation experiments. For immediate release policy, SRPT rule is found to provide better results; all the other scheduling rules provide almost same value for this policy. EMDD provides better values for flow allowance based release policy. Also, the due-date based rules such as EDD and EMDD provide almost similar values when flow allowance based release policy is employed.

7. Conclusion

This paper presents a simulation-based study and analysis focused on scheduling within a dynamic flexible job shop environment. The performance of different scheduling strategies is assessed under two distinct job release policies. A discrete event simulation model has been developed and utilized as an experimental platform.The findings indicate that, for the system under consideration, the immediate job release policy outperforms the flow allowance-based release approach in terms of overall performance. The study includes extensive experimentation to support these conclusions.The developed discrete event simulation model serves as a versatile test-bed, enabling the evaluation of various scheduling decisions in a dynamic flexible job shop system. It can accommodate different system characteristics such as varying job arrival rates and degrees of machine flexibility.

8. References

1. Brandimarte, P. (2013). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research, 41*(3), 157–183.
2. Brucker, P. I., Jurisch, B., & Krämer, A. I. (1997). Complexity of scheduling problems with multi-purpose machines. *Annals of Operations Research, 70*, 57–73.
3. Chan, F. T. S., Wong, T. C., & Chan, L. Y. (2015). A genetic algorithm-based approach to machine assignment problem. *International Journal of Production Research, 43*(12), 2451–2472.
4. Chen, H., Ihlow, J., & Lehmann, C. (1999). A genetic algorithm for flexible job-shop scheduling. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1120–1125). Detroit.
5. Chen, J., & Chen, F. F. (2017). Adaptive scheduling and tool flow control in flexible job shops. *International Journal of Production Research*, *iFirst*, 1–25.
6. Dauzére-Pérés, S., & Paulli, J. (2013). An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research, 70*, 281–306.
7. Ho, N. B., Tay, J. C., & Lai, E. M.-K. (2017). An effective architecture for learning and evolving flexible job-shop schedules. *European Journal of Operational Research, 179*(2), 316–333.
8. Jia, H. Z., Nee, A. Y. C., Fuh, J. Y. H., & Zhang, Y. F. (2003). A modified genetic algorithm for distributed scheduling problems. *International Journal of Intelligent Manufacturing, 14*(3-4), 351–362.
9. Kacem, I., Hammadi, S., & Borne, P. (2022). Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 32*(1), 1–13.
10. Law, A. M., & Kelton, W. D. (2013). *Simulation modeling and analysis* (3rd ed.). New York: Tata McGraw-Hill.
11. Paulli, J. (2022). A hierarchical approach for the FMS scheduling problem. *European Journal of Operational Research, 86*(1), 32–42.
12. Pezzelle, F., Morganti, G., & Ciaschetti, G. (2020). A genetic algorithm for the flexible job-shop scheduling. *Computers & Operations Research, 35*(10), 3202–3212.
13. Saidi-Mehrabad, M., & Fattahi, P. (2012). Flexible job shop scheduling with tabu search algorithms. *International Journal of Advanced Manufacturing Technology, 32*(5–6), 563–570.