2002

Flow Shop Scheduling Problem: a Computational Study

Amr Arisha
Technological University Dublin, amr.arisha@dit.ie

Paul Young
Dublin City University

Mohie El Baradie
Dublin City University

Recommended Citation

Arish, A., Young, P., El Baradie, M.:Flow Shop Scheduling Problem: a Computational Study.Sixth International Conference on Production Engineering and Design for Development (PEDD6),Cairo, Egypt, pp 543 – 557.

FLOW SHOP SCHEDULING PROBLEM: A COMPUTATIONAL STUDY

A. Arisha, P. Young, and M. EL Baradie

School of Mechanical & Manufacturing Engineering, Dublin City University
Dublin 9, Ireland
Email: amr.arisha@gmail.com

ABSTRACT:

A computational study has been developed to obtain optimal / near optimal solution for the flow shop scheduling problem with make-span minimization as the primary criterion and the minimization of either the mean completion time, total waiting time or total idle time as the secondary criterion. The objective is to determine a sequence of operations in which to process 'n' jobs on 'm' machines in same order (flow shop environment) where skipping is allowed. The Simulation approach for deterministic and stochastic flow shop scheduling has been developed. It reads and manipulates data for 500 jobs on 500 machines. Different factorial experiments present a comparative study on the performance of different dispatching rules, such as FCFS, SPT, LPT, SRPT and LRPT with respect to the objectives of minimizing makespan, mean flow time, waiting time of jobs, and idle time of machines.
The proposed model is evaluated and found to be relatively more effective in finding optimal/ near optimal solutions in many cases. The influence of the problem size in computational time for this model is discussed and recommendations for further research are presented.

KEYWORDS: Flow Shop Scheduling, Simulation, Dispatching Rules, and Enumerative Optimization.

1. INTRODUCTION

The purpose of this paper is twofold: (1) to provide a simulation program able to find the optimum / near optimum sequence for general flow shop scheduling problem with make-span minimization as main criteria; (2) to compare different dispatching rules on minimizing multiple criteria.
Numerous combinatorial optimization procedures have been proposed for solving the general flowshop problem with the maximum flow time criterion. Many researches have been successful in developing efficient solution algorithms for flowshop scheduling and sequencing [1, 2, 3, 4, 5, 6, 7 and 8] using up to 10 machines. Dannenbring [2] found that for small size shop problems his heuristic outperformed others in minimizing the make-span for 1280 flowshop scheduling problems. Ezat and El Baradie carried a simulation study for pure flowshop scheduling with make-span minimization as a major criterion for n ψ90 on m ψ90 [9]. In This paper study general flow shop scheduling problem with make-span minimization as main criteria for n ψ 250 and m ψ 250 with different ranges of random numbers generated (0-99) for processing times matrix.

2. THE FLOWSHOP SCHEDULING PROBLEM

The flowshop problem has interested researchers for nearly half a century. The flowshop problem consists of two major elements: (1) a production system of ‘m’ machines; and (2) a set of ‘n’ jobs to be processed on these machines. All ‘n’ jobs are so similar that they have essential the same order of processing on the M machines, Fig. 1. The focus of this problem is to sequence or order the ‘n’ jobs through the ‘m’ machine(s) production system so that some measure of production cost is minimized [10]. Indeed, flowshop scheduling problem has been shown to be NP-complete for non-preemptive schedules [11].

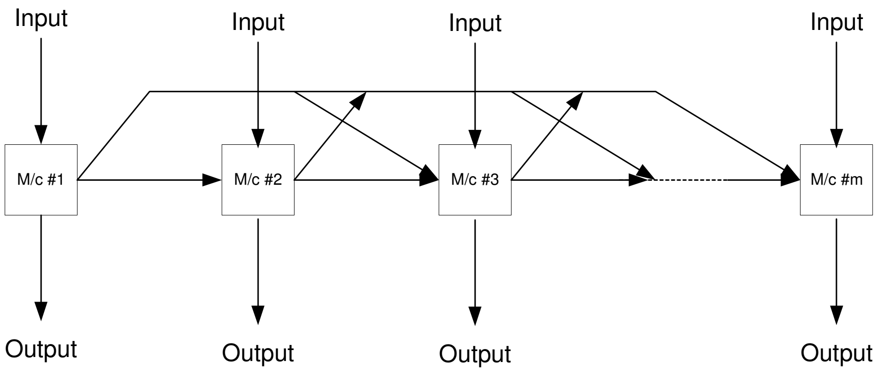


Fig.1: Work flow in General Flow Shop Scheduling Model

The assumptions of the flowshop problem are well documented in the production research literature [3,4,5,18]. In summary:

1) All ‘n’ jobs are available for processing, beginning on machine1, at time zero.
2) Once started into the process, one job may not pass another, but must remain in the same sequence position for its entire processing through the ‘m’ machines.
3) Each job may be processed on only a single machine at one time, so that job splitting is not permitted.
4) There is only one of each type of machine available.
5) At most, only one job at a time can be processed on an individual machine.
6) The processing times of all ‘n’ jobs on each of the ‘m’ machines are predetermined.
7) The set-up times for the jobs are sequence independent so that set-up times can be considered a part of the processing times.
8) In-process inventory is allowed between consecutive machines in the production system.
9) Non-preemption; whereas operations can not be interrupted and each machine can handle only one job at a time.
10) Skipping is allowed in this model.

3. THE PERFORMANCE CRITERIA

The performance criteria are those most commonly used as proposed by Stafford [15], for optimizing the general flowshop model.

1. Makespan
   Throughout the half century of flowshop scheduling research, the predominant objective function has been to minimize make-span. [10]
   The expression used is as follows:

$$\text{Minimize: } C_{max}$$

2. Mean Completion Time
   Conway et al. (1967), Panwalker and Khan (1975), Bensal (1977), and Scwarc (1983) have all discussed mean job completion time or mean flow time as an appropriate measure of the quality of a flowshop scheduling problem solution. Mean job completion time may be expressed as follows:

$$\overline{C} = \sum_{i=1}^{n} \text{ Job completion times } / \text{ n}$$

3. Total Waiting Time
   Minimizing total job idle time, while the jobs wait for the next machine in the processing sequence to be ready to process them, may be expressed as follows:

$$W_{(nxm)} = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}$$

4. Total Idle Time
   Overall all machine idle time will be considered in this model (the time that machines 2,…. , M spend waiting for the first job in the sequence to arrive will be counted). Overall machine idle time may be minimized according to the following expression:

$$\text{Minimize: } \sum_{1=1}^{m}\sum_{j=1}^{n} X_{ij}$$

4. DISPATCHING RULES

A dispatching rule is used to select the next job to be processed from a set of jobs awaiting service at a facility that becomes free. The difficulty of the choice of a dispatching rule arises from the fact that there are n! ways of sequencing 'n' jobs waiting in the queue at a particular facility and the shop floor conditions elsewhere in the shop may influence the optimal sequence of jobs at the present facility [12].
Five basic dispatching rules have been selected to be investigated in this research. A brief description about each rule will be presented:

- <u>Rule (1) FCFS (First Come First Served)</u>: This rule dispatches jobs based on their arrival times or release dates. The job that has been waiting in queue the longest is selected. The FCFS rule is simple to implement and has a number of noteworthy properties. For example, if the processing times of the jobs are random variables from the same distribution, then the FCFS rule minimizes the variance of the average waiting time. This rule tends to construct schedules that exhibit a low variance in the average total time spent by the jobs in this shop.

- <u>Rule (2) SPT (Shortest Processing Time)</u>: The SPT first rule is a widely used dispatching rule. The SPT rule minimizes the sum of the completion times ΣC_j (usually referred as the flow time), the number of jobs in the system at any point in time, and the average number of jobs in the system over time for the following machine environments: set of unique machines in series, the bank of identical machines in parallel, and the proportionate flow shop.

- <u>Rule (3) LPT (Longest Processing Time)</u>: The LPT rule is particularly useful in the case of a bank of parallel machines where the make-span has to be minimized. This rule selects the job with the longest processing (from the queue of jobs) to go next when a machine becomes available. Inherently, the LPT rule has a load balancing property, as it tends to avoid the situation where one long job is in process while all other machines are free. Therefore, after using the LPT rule to partition the jobs among the machines, it is possible to resequence the jobs for the individual machines to optimize another objective besides make-span. This rule is more effective when preemption is allowed.

- Rule (4) SRPT (Shortest Remaining Processing Time): The SRPT is a variation of SPT that is applicable when the jobs have different release dates. SRPT rule selects operations that belong to the job with the smallest total processing time remaining. It can be effective in minimizing the make-span when preemption is allowed.

- Rule (5) LRPT (longest Remaining Processing Time): The LRPT is a variation of LPT that selects the operations that belong to the job with the largest total processing time remaining. LRPT rule is of importance when preemption is allowed and especially in parallel identical machines. LRPT rule always minimizes the idle time of machines.

5. EXPERIMENTAL DESIGN

In flowshop sequencing research, the standard approach for evaluating a new problem solving technique, whether it is a heuristic or an optimization model, is to generate a set of problems of different sizes, and then to solve this common set of problems with the new techniques and with one or more other proven techniques designed for the same flowshop problem.

A computer simulation program Table. 1 has been developed into two phases: (1) to find the optimum/near optimum solution for general flowshop problem to minimize the makespan; (2) to measure the effectiveness of various priority rules for flow shop scheduling. The program can read data up to 500 x 500 and use both deterministic and stochastic processing time input. The input processing times may be generated from different seed random numbers for each single run or read directly data from an input file. The number of runs for each case is 300 where the results turn to steady state started as shown in Fig. 2.

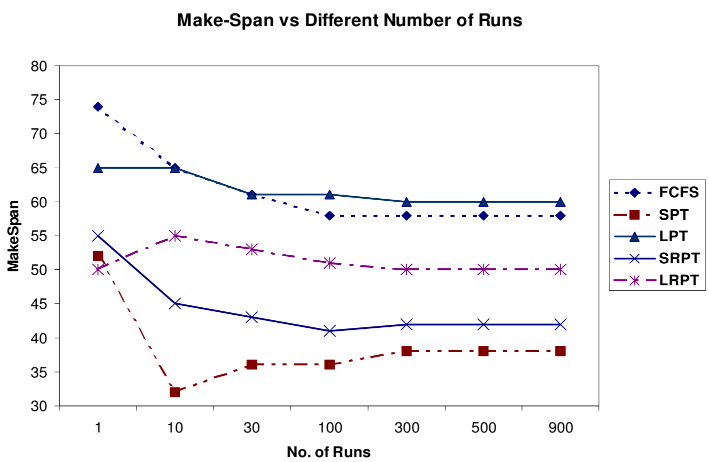


Fig.2: A Steady-State Analysis of the Model

Different factorial sets of experiments were conducted to verify that the program would provide optimal solutions to general flowshop problems and to compare between various dispatching rules.

6. EXPERIMENTAL RESULTS

Phase 1:

This phase has multi-objectives for $n/m/F/C_{max}$ problem. It can provide the followings:

1) All the job sequences and their correspondent makespan for each sequence.
2) The optimal job sequence and its makespan value.
3) Frequencies for all job sequences.
4) CPU time for the solution.

A sample of the output of the program and optimal makespan are shown in Fig. 3. Although, many researchers have been working on the flowshop sequencing problem for many years, it has been found nowhere any results about the distribution of the objective function and the distribution of the optima of this function. In effect, such an approach gives an intuitive idea about the problem and is important to allow the reader to judge the quality of the method used for this problem [13]. The distribution of all the possible make-spans obtained by complete enumeration of two different problems are given in Fig. 4. This distribution is given relative to the optimum solution. The processing times were randomly generated (integers between 1 and 10). The distribution seems to be almost symmetric and its range is contained in an interval of 20% around the mean. A χ^2 test does neither confirm nor refute that this distribution is Gaussian, therefore the use of the mean makespan given by a heuristic seems to be meaningful.

Table 1: Terminology associated with Simulation Model

Model parameters

Model $n/m/F/C_{max}$

- n — number of jobs to be processed,
- m — number of machines (processing steps),
- F — general flowshop scheduling problem,
- C_{max} — the criterion is Make-Span,
- P_{ij} — processing time off job 'j' on machine 'i',
- $\overline{C}$ — the criterion is Mean Completion Time,
- w_{ij} — waiting time before start the job 'j',
- X_{ij} — idle time of machine 'i' before start job in position j in the sequence,
- $W_{(nxm)}$ — total waiting time.

Dispatching Rules

FCFS : First Come First Served
SPT : Shortest Processing Time
LPT : Longest Processing Time
SRPT : Shortest Remaining Processing Time
LRPT : Longest Remaining Processing Time

Decision Variables

$n > 0$ $i = 1, \ldots\ldots, n$,
$n < 250$ (recommended) (the model can read up to 500 jobs),
$m > 0$ $j = 1, \ldots\ldots, m$,
$m < 250$ (recommended) (the model can read up to 500 machines),
Number of runs,
Number of seed,

Objective functions

1) Make-span
Minimize: C_{max}

2) Mean Completion Time
Minimize: $\overline{C} = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{m}(w_{ij} + P_{ij})$

3) Total Waiting Time
Minimize: $W_{(nxm)} = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}$

4) Total Idle Time
Minimize: $\sum_{1=1}^{m}\sum_{j=1}^{n} X_{ij}$

```
5 jobs on 7 machines

   n\ m  M1   M2   M3   M4   M5   M6   M7
   J1    8    9    7    8    6    2    8
   J2    3    5    9    7    5    2    4
   J3    9    1    8    0    10   8    2
   J4    3    8    10   2    3    0    0
   J5    4    2    6    8    10   1    0

5 Jobs, 7 Machines, Random seed=48
    Number of runs           = 1

12345 Make-Span=75
12354 Make-Span=75
12435 Make-Span=75
24153 Make-Span=71
24513 Make-Span=72
24531 Make-Span=67
31524 Make-Span=66
34125 Make-Span=66
34152 Make-Span=64
41532 Make-Span=66
45312 Make-Span=61
45321 Make-Span=63
54132 Make-Span=66
51342 Make-Span=64
51432 Make-Span=66
. . . .
. . . . .
Number of Job Sequences = 120

53421  = Optimum Job Sequence, Optimal Make-span = 60
CPU time in seconds =0.211000

Make-Span's Frequencies:

For the Make-Span =60: the Frequency =3
For the Make-Span =61: the Frequency =3
For the Make-Span =62: the Frequency =1
For the Make-Span =63: the Frequency =3
For the Make-Span =64: the Frequency =15
For the Make-Span =66: the Frequency =15
For the Make-Span =67: the Frequency =20
For the Make-Span =68: the Frequency =6
For the Make-Span =70: the Frequency =1
For the Make-Span =71: the Frequency =39
For the Make-Span =72: the Frequency =1
For the Make-Span =73: the Frequency =3
For the Make-Span =75: the Frequency =10
53421= Optimum Job Sequence, Optimal Make-Span = 60
```

Input matrix of processing times

Job sequences

The entire possible job sequences

The frequency of each make-span

Optimal Job Sequence

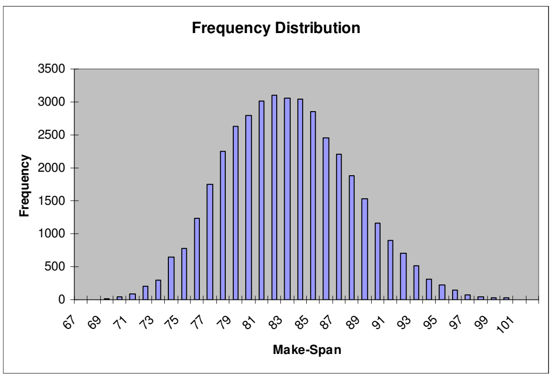
Fig. 3: Sample of program output for Phase 1

```
8 Jobs, 8 Machines, Random seed=33
  Number of runs             = 1
 Number of Job Sequences = 40320

  6,5,4,3,8,7,1,2  = Optimum Job
 Sequence, Optimal Make-span = 67
  CPU time in seconds =61.519000
```

```
10 Jobs, 30 Machines, Random seed=25
  Number of runs             = 1
 Number of Job Sequences = 3628800

  8,5,10,7,6,4,9,3,2,1  = Optimum Job
  Sequence, Optimal Make-span = 109
 CPU time in seconds =1168.229000
```



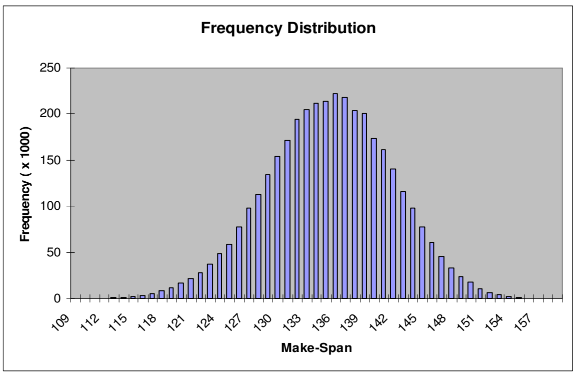


Fig. 4: Frequency distribution of two different flowshop problems

Phase 2:

To compare the dispatching rules under identical conditions, the same pseudo random numbers generated per run, and the same number of runs. A different factorial experiment for the selected rules, 7 machines shop (5, 20, 50, 80, 130, 200, 250), and 9 levels of Work In Process (WIP) (number of jobs in shop) equal to (5, 10, 30, 50, 80, 100, 150, 200, 250) were executed. Computational times for the entire experiments are shown in Table. 4. Samples of machine shop were presented for each performance criteria.

- **Average Make-Span Criterion**

  For small machine numbers, Fig. 5, there is a clear spread across the different rules with SPT rule providing the best results and the LPT rule performing worst.
  For larger machine numbers, Fig. 6, the LPT rule is still clearly the worst, however the other rules show almost identical results. Nevertheless the SPT rule is the best performer overall.

- **Average Mean Completion Time Criterion**

  For this criterion, Fig. 7, the SRPT rule provides the best results. Again, the LPT rule performs worst, sometimes rivaled by the LRPT rule.

- **Average Waiting Time Criterion**

  This criterion changes the order of the rules with LPT performing best and SPT performing worst, Fig. 8. In addition increasing the number of jobs increases the spread between best and worst results significantly.

- **Average Idle Time Criterion**

  As with the waiting time criterion, Fig. 9 shows that the spread of the performance increases with job number. Here the LRPT rule is clearly the best while the LPT rule performs worst.

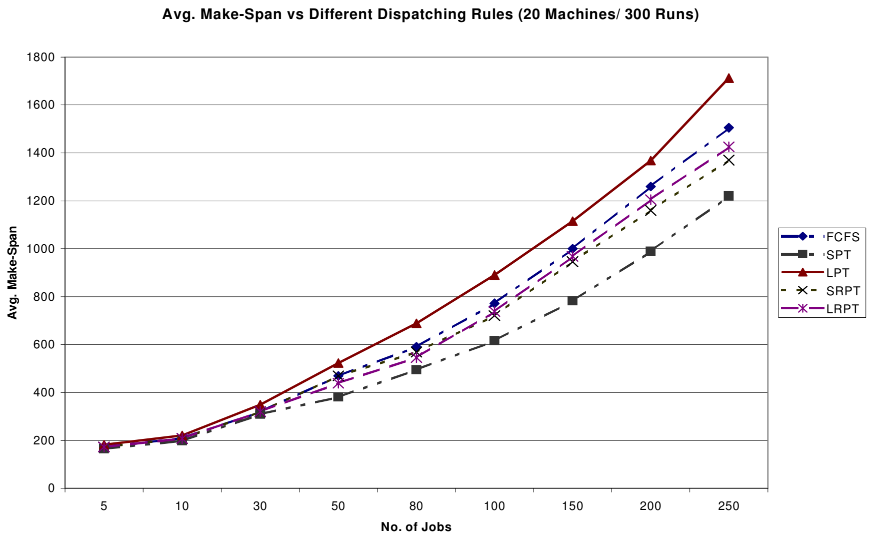


Fig. 5: Effect of the dispatching rules vs. average makespan under different machine shop

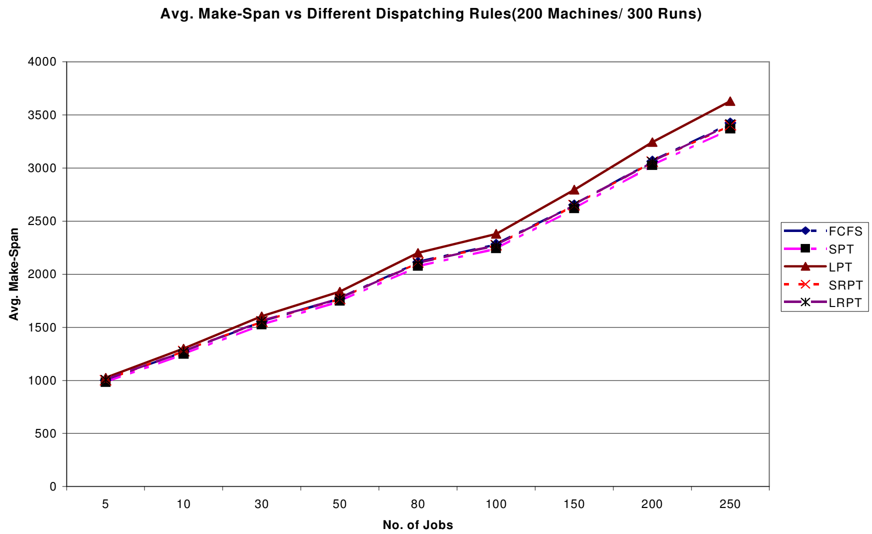


Fig. 6: Effect of the dispatching rules vs. average makespan under different machine shop

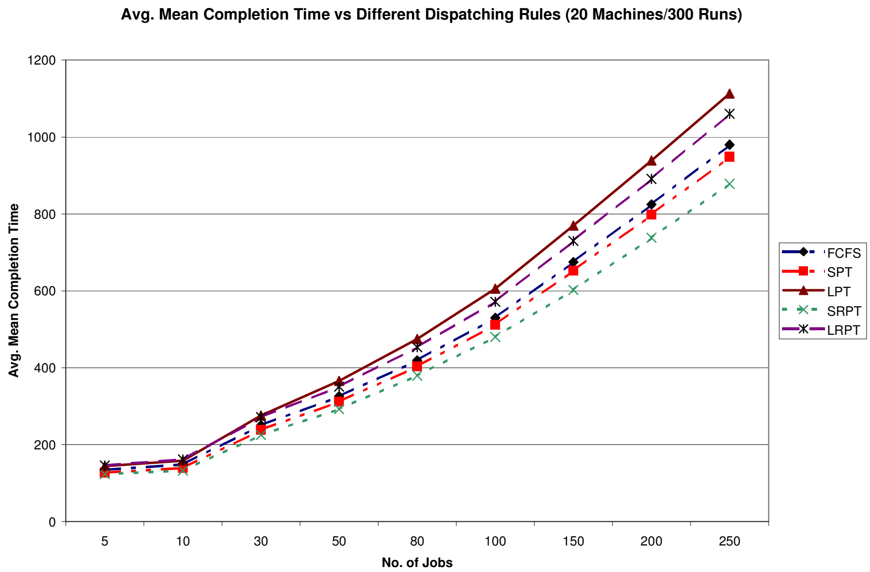


Fig. 7: Effect of the dispatching rules vs. average mean completion time under different machine shop

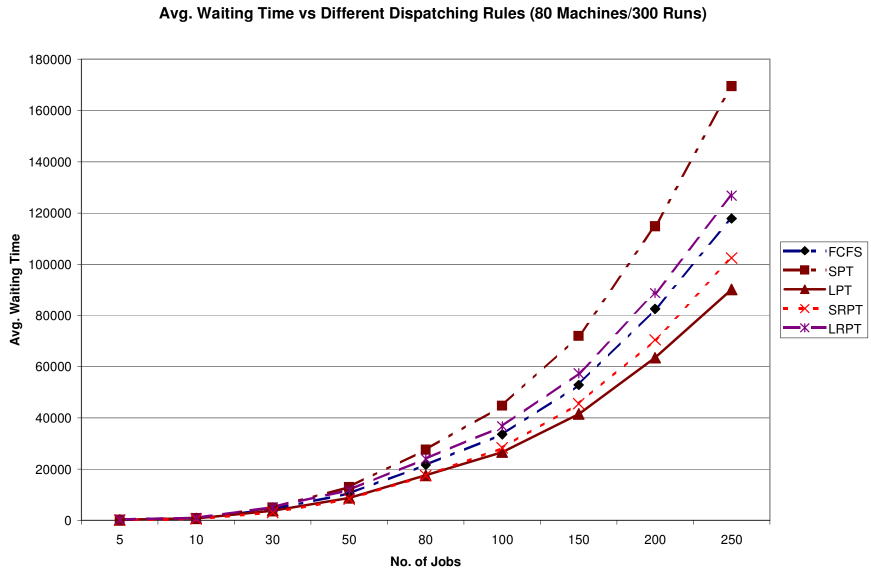


Fig. 8: Effect of the dispatching rules vs. average waiting time under different machine shop

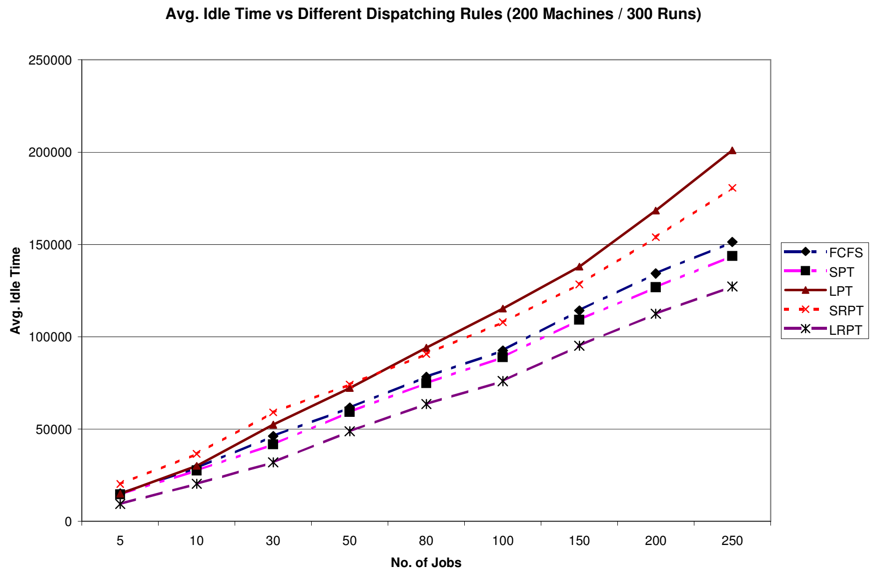


Fig. 9: Effect of the dispatching rules vs. average idle time under different machine shop

7. RESULTS ANALYSIS

In general, the quality of a technique's solutions is measured in at least two dimensions: (1) how close the solution comes to the optimal solution if it can be measured; and (2) how much computer time is required to solve problems of a given size.

Due to wide differences in software, platform, problem size, experimental design and reporting, it is very difficult to compare the performance of different techniques directly. To allow some comparison to be made Table 2 shows the average percentage increase over optimum make-span time as reported by each of the researchers for their algorithms. To enhance the comparison, the right column indicates the relative limitations of each model.

Table 2: Comparison between some different studies

Method	Avg. % of increased of Optimum	Limitations
Palmer, 1965	10 % - 35 %	Optimum achieved in 30% of cases Small scale problems only
Campbell, 1970	5 % – 20 %	Optimum is not guaranteed Economical n < 8 ,
Dannenbring, 1977	5 % – 15 %	Optimum achieved in 35% of cases n < 6 , m < 10 only
Gupta, 1971	10 % - 20 %	Optimum is not guaranteed
Al-Qattan, 1990	0 % - 15 %	Optimum is not guaranteed
Ezat – El Baradie, 1993	0 % - 10 %	Optimum is for n < 12 m < 60 Pure flowshop scheduling problems only Max. size n < 90 , m < 90
Tsang – Stafford, 2001	0 % - 5 %	Optimum is guaranteed for n < 7 , m < 7
LEKIN, 1998 [14]	0 % - 10 %	Optimum is not guaranteed Max. size n < 10 , m < 18
Arisha – El Baradie, 2001	0 % 0 % - 10 %	Optimum is guaranteed for n < 50 , m < 250 General flowshop scheduling problems only Max. size n < 500 , m < 500 For n > 50, m > 250

The exponential increase in solution time with number of jobs is shown in Table 3 for the phase 1 simulation model used in this work. The length of time for computation means that the full search cannot be economically used where the number of jobs exceeds 30. The effect of machine numbers on time is clearly much less significant.

Table 3: CPU time (seconds) to find optimum makespan for different problem sizes

n x m	5	10	20	40	100	250
5	0.161	0.18	0.17	0.181	0.21	0.22
6	1.1	1.1	1.2	1.2	1.3	1.3
7	8.1	8.2	8.4	8.4	8.8	9.0
8	40.3	43.3	44.9	45.6	44.3	45.2
9	232.1	232.6	245.0	235.3	245.1	248.1
10	1158.0	1176.3	1256.4	1307.3	1354.2	1298.3
11	13115.3	13118.3	14234.7	13215.2	13684.3	13968.3
12	56025.3	56036.9	57231.3	56016.7	57863.3	58015.2
20	-	551369.4	-	543652.4	-	-
30	-	-	-	2605248.0	-	-

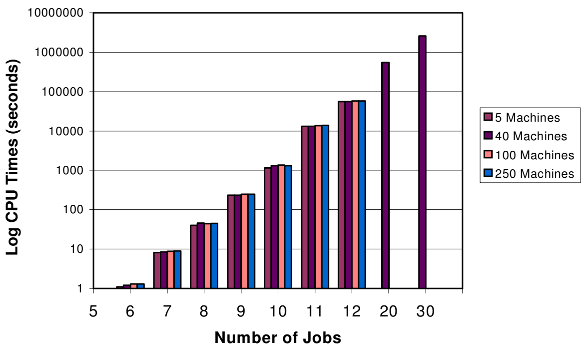


Fig.10: CPU Times vs. different Number of Jobs

Table 4: The factorial experiment for phase 2 and the CPU times

		Number of Machines						
	n x m	5	20	50	80	130	200	250
Number of Jobs	5	17.25	17.9	18.05	18.56	19.6	20.25	21.82
	10	18.45	18.96	19.5	19.91	20.45	21.68	22.35
	30	19.48	19.9	20.45	21.34	21.98	22.57	23.01
	50	20.01	20.45	20.98	21.54	21.87	22.15	23.24
	80	21.14	21.29	21.76	22.14	22.56	22.89	23.96
	100	21.82	22.15	22.6	23.04	23.42	24.12	24.54
	150	22.15	22.35	22.98	23.45	23.87	24.36	24.95
	200	22.59	22.84	23.15	23.68	24.15	24.59	25.15
	250	23.15	23.59	23.96	24.15	24.93	25.09	25.96

8. DISCUSSION

Flowshop scheduling is one of the most critical activities for the production planner. Minimizing the make-span, mean flow time, job waiting times and machine idle time are the major objectives to reduce the processing costs. The flowshop scheduling problem is NP-hard and several studies have been done to solve small size flowshop problems.

The parameters that affect the size of flowshop problems are ‘n’ and ‘m’. The problem size complexity is based on these two parameters. The results of the proposed model indicated that ‘n’ has much stronger influence on computer solution time than ‘m’. Based on these studies and the proven NP-completeness of the problem, it is clear that ‘n’ is a much more important determinant of computer solution time required for the flowshop problem.

The simulation study has been carried out under several operational conditions. It found the optimum/ near optimum solution in a reasonable computational time for most cases in a specific range of problem sizes. It is recommended to use the model for the problems where number of jobs is less than 30 and number of machines is less than 250 as it is not economical for larger scale.

Furthermore, a comparative study on the performance of various dispatching rules has been carried out under different shop machine and utilization levels. The model runs for 300 iteration using the random generator for the same specific shop conditions. It has been observed that no single rule performs well for all important criteria related to completion time, waiting time and idle time. SPT has performed the best to minimize make-span under different conditions, as is clearly evident from Fig. 11. The SPT rule is quite often used as benchmark since it is found to be very effective in minimizing make-span and also mean tardiness under highly loaded shop floor conditions [12]. SPT show the worst for job waiting criterion. While LPT shows the worst performance for make-span criterion, it tends to be the best rule to minimize the job waiting time especially for high utilization level (n > 80). For the average mean completion time (mean flow time) SRPT shows the best performance for different levels of utilization. LRPT and LPT performed worst for average mean completion time criterion. LRPT rule tends to dominate with respect to the machine idle time while LPT and SRPT showed the worst performance depending on job numbers.

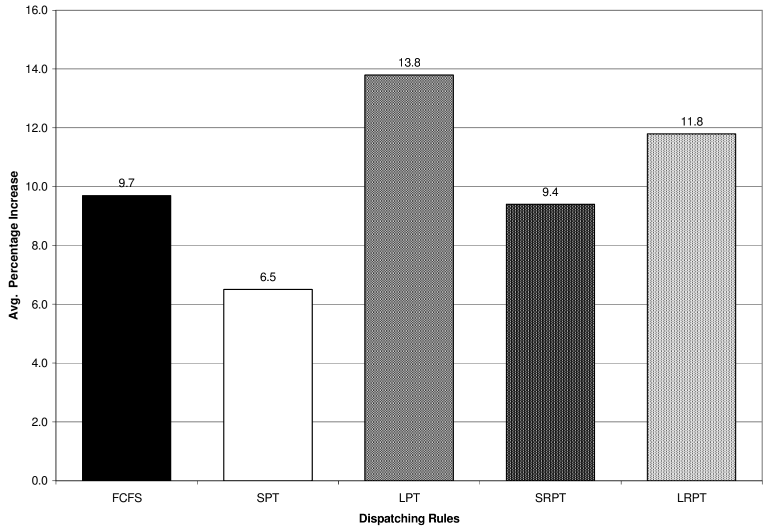


Fig. 11: Comparison of average normalized increase over optimum make-span for different dispatching rules

While never being the best or worst performer for any criterion the FCFS rule is effective in minimizing the maximum flow time and the variance in flow time. Its consistent "mid-table" performance allows its use as a benchmark.

9. CONCLUSIONS

Some recommended trends for flowshop scheduling problem are listed down:

- A full solution search is uneconomic for larger sized problems so it is important to develop intelligent search techniques that truncate the search tree size to get the optimum / near optimum solution.
- Although hundreds of publications and studies had been done in flowshop scheduling problems, but the need to investigate problem size beyond the small sizes (n < 9, m < 9) is still required [10].
- Artificial Intelligent techniques turns to be one of the most effective tools to handle optimization problem to get a satisfying solution [17].
- However, there are still many areas where more research is needed, including the integration of the three common approaches; operations research-based, simulation-based and AI-based in order to develop a comprehensive hybrid model to solve flowshop scheduling problems.

References

1. JOHNSON, S. M., "Optimal Two and Three Stage Production Schedules with Set-up time Included". Nav. Res. Log. Quart. , Vol. 1, pp. 61-68, (1954).
2. CAMPBELL, H. G. " A heuristic algorithm for n-job and m-machines sequencing problem". Management Science, Vol. 16, pp. 630-673, (1970).
3. BAKER, KENNETH R., " Sequencing and Scheduling". John Wiley & Sons Inc.(1998), ISBN: 0-9639746-1-0.
4. GUPTA, J.N.D. , "A functional heuristics algorithm for the flow-shop scheduling problem". Operation Research, Vol. 22, No. 1, pp.39- 48, (1971).
5. FRENCH, S. "Sequencing and Scheduling". An introduction to the mathematics of the job shop, Ellis Horwood series, (1982).
6. DANNENBRING, D. G. , " An evaluation of flowshop sequencing heuristics". Management Science, Vol. 23, No. 11, pp. 1174-1182, (1977).
7. PINEDO, MICHEAL and BHASKARAN, K. " Dispatching". Handbook of Industrial Engineering, pp. 2182-2198, (1992).
8. BERA, H., "To minimize waiting time of jobs, idle time of machines and total elapsed time for N jobs on M machines". 2nd Joint international conference on Production Engineering, Leicester Polytechnis, pp. 290-299, (1983).
9. EZAT, A. M. and EL BARADIE, M. " Flow-Shop Scheduling: Effect of Various Priority Rules on Minimizing Multiple Criteria". 30th International MATADOR Conference (UMIST), (1993).
10. TSENG, FAN and STAFFORD, E. ,"Two MILP models for N x M SDST flowshop sequencing problem ". Int. J. Prod. Res., Vol. 39, No. 8, 1777 – 1809, (2001).
11. GRAVES, STEPHEN ," A Review of Production Scheduling". Operations Research, Vol. 29, pp. 646-675, (1981).
12. RAJENDRAN, C. and HOLTHAUS, OLIVER, " A Comparative study of dispatching rules in dynamic flowshops and jobshops". European Journal of Operational Research, Vol. 116, pp.156-170, (1999).
13. TAILLARD, ERIC D., " Some efficient heuristic methods for the flowshop sequencing problem". European Journal of Operational Research, Vol.47, pp. 65-74, (1990).
14. PINEDO, MICHEAL, " Operations scheduling with applications in manufacturing and services". Irwin/McGraw-Hill, (1998). ISBN: 0072897791.
15. STAFFORD, E. F., "On the development of a mixed-integer linear programming model for the standard flowshop". Journal of the Operational Research Society, Vol. 39, pp. 1163-1174, (1988).
16. BLAZEWICZ, J., ECKER, KLAUS H., SCHMIDT, GUENTER and WEGLARZ, J., "Scheduling in Computer and Manufacturing Systems". Springer Verlag, Berlin (1994). ISBN: 3540580492.
17. MEZIANE, VADERA, KOBBACY and PROUDLOVE, "Intelligent Systems in Manufacturing: Current Developments and Future Prospects". Integrated Manufacturing System, Vol. 11, No. 4, pp. 218-238, (2000).
18. CONWAY, R. W., MAXWELL, W.L. and MILLER, I. W. ," Theory of Scheduling ". Addison Wesley, Reading, Massachusetts, (1967).