EFFECT OF VARYING LEVELS OF SETUP TIME ON MAKESPAN CRITERIA: A COMPARATIVE STUDY OF TWO ALGORITHMS ON SDST FLOW SHOP

Rajesh Vanchipura[1] & Sridharan R.[2]

Abstract: *This paper presents a comparative study of two algorithms for the flow shop scheduling problem with sequence dependent setup times. The objective of the study is to investigate the effect of setup time on makespan under varying proportions of setup time. The two constructive algorithms considered in the study differ in their approach in constructing the sequence; the first algorithm is processing time based and the second algorithm is setup time and processing time based. Taillard benchmark flow shop problems are used for development of benchmark problems for flow shop with sequence dependent setup time. The two algorithms are tested for the 960 problem instances developed. The results are analyzed graphically and statistically for the varying levels of setup time. The study reveals that the setup time based algorithm improves its performance as the setup time increases as percentage of processing time. This paper also suggests a framework namely setup dominance matrix, which provides the flexibility in identifying the algorithm to be used for a specific size flow shop scheduling problem at a particular level of setup time.*

Keywords: Flow shop, sequence dependent setup time, heuristic algorithm, makespan.

[1] Department of Mechanical Engineering, Govt. Engineering College, Thrissur - 680 009, Kerala, India.,rajeshvanchipura@gmail.com
[2] Department of Mechanical Engineering,National Institute of Technology Calicut, Kerala -673 601, India., sreedhar@nitc.ac.in

1. Introduction

The flow shop problem with sequence dependent setup time has been addressed by many researchers in the last few decades. This problem involves scheduling a set of *n* jobs which are available for processing on *m* machines. The setup time of each job is sequence dependent i.e., the setup time of each job on a machine is dependent on the previous job on the same machine. The problem is to determine the schedule which minimizes the makespan assuming no pre-emption of operations. The problem is commonly known as Sequence Dependent Setup Time (SDST) flow shop scheduling problem. The problem finds relevance in many areas such as textile, chemical, pharmaceutical, food processing, metal processing, paper manufacturing, and many other industries (Eren, 2011). But, in many situations, the setup times are conveniently added to processing time by compromising on the quality of solution. The presence of sequence dependent setup time will certainly make the problem solution more complex. But, its benefits can be seen at the production floor by means of accurate estimates of completion time and accurate delivery time promises. The SDST flow shop scheduling problem falls in the category of NP-complete (Gupta, 1986). Owing to the complexity, many of the research works consider only two machines configurations such as Hwang and Sun (1998), Sun and Hwang (2001), and Mansouri (2009). Heuristics methods are the popular choice of most of the research works carried out in this field. Rios-Mercado and Bard (1998) present two heuristics for the SDST flow shop scheduling problem with makespan objective. Simons (1992) develops four heuristic algorithms and compares them with three benchmarks that represent generally adopted approaches to scheduling in this environment. Ruiz et al. (2005) propose two heuristics based on genetic algorithm for SDST flow shop. Gajpal et al. (2006) present an ant colony optimization algorithm for flow shop scheduling with sequence dependent setups for the makespan objective. The literature shows that metaheuristics have emerged as a popular methodology in the last decade. Many authors such as Wang et al. (2011), Naderi et al. (2009), Naderi et al. (2010), Gholami et al. (2009), Salmasi et al. (2009) and Rajendran and Ziegler (2003) use metaheuristics for SDST flow shop scheduling .

In a flow shop operating in a sequence dependent setup time environment, the processing times and setup times vary from industry to industry. The performance of various algorithms used to solve the problem may depend on the process time and setup time data. For the purpose of normalisation, average setup time can be expressed as a proportion of processing time. The proportion of setup time to processing time also can vary with situations. This variation of setup time will have an effect on the performance of different algorithms used.

To the best knowledge of the authors, the effect setup time variation on the performance of different SDST flow shop scheduling algorithms is an area unexplored by the researchers. The present paper addresses this particular research gap. The present study considers two algorithms (a setup time based algorithm and a processing time based algorithm) for detailed investigation. A total of 960 SDST flow shop problem instances have been solved. Relative performance improvement analysis and statistical analysis are performed on the results.

2. Formulation of the flow shop scheduling problem with sequence dependent setup times

A flow shop scheduling problem involves a set of n jobs to be processed on a set of m machines in the same order. For a general flow shop, the job sequence may not be the same for every machine. But, in this paper, the assumption is permutation schedule which considers same sequence for all machines. It is also assumed that setup time is sequence dependent i.e., the setup time of a job on a machine is determined only by knowing the predecessor of the job. All the assumptions for a static flow shop is valid for the SDST flow shop. It is also assumed that pre-emption is not allowed. The objective of this problem is to minimise the time at which the last job in the sequence finishes processing on its last machine, i.e. minimise the makespan.

Notations

n Total number of jobs to be scheduled

i Index of machine; (i-1) indicates the previous machine in the sequence

m Total number of machines in the flow shop

j Index of job, (j-1) indicates the previous job processed in the machine

p_{ij} Processing time of job j on machine i

s_{ijk} Setup time on machine i, when job k is preceded by the job j

σ Ordered set of jobs already scheduled out of n jobs; partial sequence

$q(\sigma, i)$ Completion time of partial sequence σ on machine i (i.e. the release time of machine i after processing all jobs in the partial sequence σ)

$q(\sigma j, i)$ Completion time of job j on machine i, when job j is appended to the partial sequence σ

For calculating the start and completion times of jobs on machines in the permutation flow shop, recursive equations are used as follows.

The completion time of σj on machine i is determined using the following recursive equation:

$$q(\sigma j, i) = max\{q(\sigma, i) + s_{ijk}, q(\sigma j, i-1)\} + p_{ij}$$

where q $(\Phi, i) = 0$ and $q(\sigma, 0) = 0$, for all σ and i, with Φ denoting a null schedule. It is assumed that s_{ijk} exists for all jobs where $j = \Phi$ for all machines. It is also assumed that setup of a machine can be done without the job being available at the machine.

The flow time of job j, C_j, is given by

$$C_j = q(\sigma j, m)$$

When all the jobs are scheduled, the makespan M is obtained as follows:

$$M = max\{C_j, j = 1, 2, \dots, n\}$$

3. Solution methodology

Heuristic algorithms are predominantly used for SDST flow shop scheduling problems because of the complexity involved in solving the problem. The general flow shop scheduling problem is NP hard and the presence of sequence dependent setup time makes it further hard. This paper addresses two constructive algorithms NEHRB heuristic (the existing algorithm) and Fictitious Job Setup Ranking Algorithm, FJSRA (the proposed algorithm). The first algorithm is processing time based which considers only processing time in the construction of sequence and the second algorithm is setup time based. These algorithms are described briefly in the following sections.

3.1 Fictitious job setup ranking algorithm

Fictitious Job Setup Ranking Algorithm (FJSRA) is developed by giving importance to setup time. Initially setup time matrices of all the machines are summed to get summed setup time matrix, SST_{ij}. This SST_{ij} is then sorted based on the setup time values to get the Sorted Summed Setup Time matrix, $SSST_{ij}$. The $SSST_{ij}$ shows the best combinations of jobs in terms of setup time i.e., the jobs having the lowest setup time between them. The best combinations of jobs are considered as the fictitious jobs. Insertion procedures are carried out on these fictitious jobs to get the final sequence. The advantage of this algorithm is that it does not disrupt the best combinations of jobs.

3.2 NEHRB heuristic algorithm

This heuristic is based on NEH heuristic (Nawaz et al., 1983) for flow shop scheduling problem. The procedure adopted in NEH heuristic consists of inserting a job into the best available position of the partial sequence. Rios-Mercado and Bard (1998) extend the NEH heuristic to solve the sequence dependent setup flow shop problem. This procedure is known as NEHRB heuristic. This algorithm uses multiple iterations in which there is a partial schedule in each iteration. The largest processing time (LPT) rule is applied to select the next job to be inserted into the partial sequence. The makespan is compared at each stage and the best sequence is selected. This procedure is continued till the final sequence is obtained.

4. Experimentation

For the purpose of experimentation, benchmark problems are developed for flow shop scheduling problems with SDST. Taillard benchmark problems for general flow shop problems are used to develop the required set of problems. Taillard benchmark has 12 different sizes of problems. All these problems are converted to SDST flow shop scheduling problem instances by augmenting setup time matrices for each machine in the benchmark problem (Table 1).

Table 1: SDST flow shop benchmark problems derived using Taillard problems

Problem size	Instances
SDST Benchmark 5	120
SDST Benchmark 10	120
SDST Benchmark 25	120
SDST Benchmark 50	120
SDST Benchmark 75	120
SDST Benchmark 100	120
SDST Benchmark 125	120
SDST Benchmark 150	120
Total	960

For each size problem, eight different proportions of maximum setup time to maximum processing time are used. These proportions are 5%, 10%, 25%, 50%, 75%, 100%, 125% and 150%. Corresponding to these proportions, setup times are generated using uniform distribution U (1, 5), U (1, 10), U (1, 25), U (1, 50), U (1, 75), U (1, 100), U (1, 125) and U (1, 150) respectively. The eight different sets of benchmark problems generated are shown in Table 1. A total of 960 problem instances are developed with 12 sizes of problems, at eight different levels of setup time forming 96 groups of problems with 10 instances in each. The two algorithms considered here are coded in MATLAB and applied for solving all the 960 problem instances developed.

5. Results and Discussion

The makespan results obtained are grouped into 12 groups, where each group has one problem size and its eight setup time variations. The objective of the study is to analyse the effect of varying setup time on the performance of the two algorithms namely, NEHRB heuristic and FJSRA heuristic. Analysing the makespan results obtained, it is observed that even in a single size problem, the makespan values are found to vary drastically with change in the proportions of setup time. Hence, it is necessary to normalise the makespan values for the purpose of comparison. This is done by determining the Relative Performance Improvement (RPI) for each problem instances. Since RPI is the percentage improvement in makespan, it can be used to compare groups of problems which are having entirely different makespan values. The The RPI used for the comparison of the heuristic algorithms is computed as follows.

$$\text{Relative Performance Improvement} = \frac{NEHRB\ C_{max} - FJSRA\ C_{max}}{NEHRB\ C_{max}}$$

where $NEHRB\ C_{max}$ is the makespan found using NEHRB heuristic and $FJSRA\ C_{max}$ is the makespan found using FJSRA heuristic. The RPI is used for comparison of heuristic algorithm between groups. The RPI values are found out for each problem instances and the average RPI values for each group are calculated by taking the average the RPI values of the respective problem instances in the group.

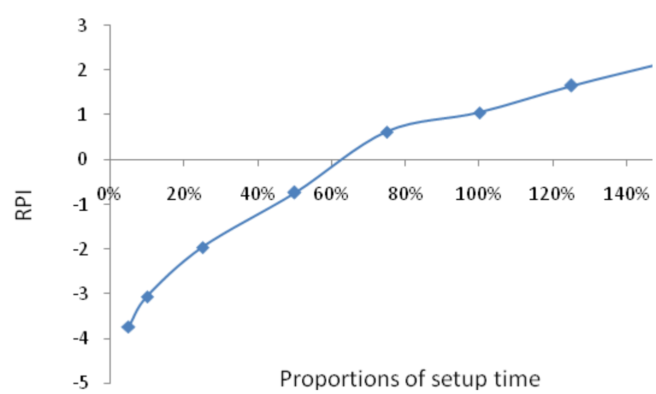

Figure 1: Average relative performance improvement for varying levels of setup time

Figure 1 presents the variation of average Relative Performance Improvement (RPI) with increasing proportions of setup time. The graph is in the negative range up to around 60% level of setup time. This indicates that at lower level of the proportions of setup time (the proportions such as 5%, 10%, 25%, and 50% levels), NEHRB heuristic clearly dominates the setup time based algorithm FJSRA. This is because the processing times are significantly higher than the setup time at these levels. The Figure 1 shows that the initial portion of graph is relatively steep which indicates a relatively high rate of increase in RPI at these levels. The above analysis is carried out for all the 12 problem sizes. However, due to space limitations, it is not included in this paper. The linearly increasing improvement is visible in all problem sizes. Further, this analysis reveals that the NEHRB heuristic is dominant for smaller size problems while FJSRA heuristic is dominant for larger size problems.

5.1 Statistical Analysis

The graphical analysis shows that the performance of algorithms for SDST flow shop varies with the proportions of setup time and problem size. The graphical analysis does not validate the superiority of any particular algorithm for a specified group of problems. Hence, a statistical analysis is carried out to obtain inferences about a specific problem size. Moreover, statistical analysis is useful in deciding the statistically superior algorithm for a particular size problem. A paired comparison *t* test with significance level $\alpha = 0.05$ is performed for all the 12 groups of problems. μ_d is the mean difference of makespan of the two algorithms, NEHRB and FJSRA. The hypotheses are formulated as follows.

Null hypothesis:

$H_{0:}$ μ_d=0 (There is no significant difference between mean makespan of the algorithms)

Alternate hypothesis:

$H_{1:}$ $\mu_d > 0$ (There is a significant difference; FJSRA has lesser makespan compared to NEHRB)

The null hypothesis H_0 is accepted when the *t* statistic is less than the critical value of *t* at significance level $\alpha = 0.05$. A higher *t* statistic value beyond the critical value will lead to rejecting the null hypothesis and accepting the alternate hypothesis. This means that the difference is significant and FJSRA is statistically superior to NEHRB.

Table 2: Results of statistical analysis for SDST benchmark 100

Problem group	Results of statistical analysis			
	Mean	*t*	*t* statistic	Result
SDST 20x 5	-179.1	1.83	-1.18533	Similar
SDST 20x	-285.1	1.83	-4.78071	Similar
SDST 20x	-350.9	1.83	-6.82233	Similar
SDST 50x 5	-149.4	1.83	4.702699	FJSRA
SDST 50x	-368.2	1.83	2.539012	FJSRA
SDST 50x	-508.2	1.83	-2.23153	Similar
SDST 100x	28.4	1.83	21.98318	FJSRA

SDST 100x	-492.4	1.83	5.576246	FJSRA
SDST 100x	-914.7	1.83	2.678399	FJSRA
SDST 200x	-102.9	1.83	20.51238	FJSRA
SDST 200x	-1023.7	1.83	8.179314	FJSRA
SDST 500x	-39.6	1.83	27.57935	FJSRA

The *t* test is carried out for all the 96 groups of problems. Due to space limitations, the details of the analysis for 12 groups of SDST benchmark 100 are presented in Table 2. The Table gives the mean difference in makespan between the two algorithms, the critical values of *t* at 0.05% significance level, the test statistic for each group and the final result of the test. The first row of the Table shows for the problem SDST 20 x 5, the *t* statistic value (-1.1853) is less than the critical value (1.83). Hence, the null hypothesis is accepted which leads to the inference that there is no statistical difference between the two algorithms. For the problem SDST 50 x 5, the *t* statistic value (4.702) is greater than the critical value (1.83). Here, the null hypothesis is rejected. This leads to the conclusion that FJSRA is better. The statistical test reveals that for eight problem sizes, FJSRA is superior and for the other four groups, there is no significant difference in performance between the two algorithms.

5.2 Development of setup time dominance matrix

From the statistical analysis, it is evident that there is no one algorithm that dominates the other algorithm for all problem sizes. This particular situation demands a framework which can be used decide the algorithm for solving a particular size of problem when proportion of setup time changes. A framework called setup time dominance matrix is developed for this purpose. The logic of this framework is as follows: For any size of the problem, as the setup time expressed as a percentage of processing time increases, FJSRA becomes superior algorithm at a certain level of setup time. The setup time dominance matrix is shown in Table 3. An entry "0" in the Table shows that there is no difference between the two algorithms and an entry "1" shows that FJSRA is better than NEHRB heuristic. The matrix is prepared based on the statistical analysis done for all the 96 groups of problem instances. The results of statistical analysis are presented on a problem size versus setup time level matrix to form the setup time dominance matrix.

Table 3: Setup dominance matrix

Problem size	Setup time as percentage of processing time							
	5%	10%	25%	50%	75%	100%	125%	150%
SDST 20 X 5	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SDST 20 X10	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SDST 20 X 20	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

SDST 50 X 5	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
SDST 50 X 10	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**
SDST 50 X 20	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SDST 100 X5	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
SDST 100 X 10	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
SDST 100 X 20	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**
SDST 500 X5	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
SDST 500 X10	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
SDST 500 X 20	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**

6. Conclusion

This paper presents the analysis of heuristic algorithms for flow shop problems with sequence dependent setup time by allowing the setup time to vary at eight different levels. The major finding from relative performance improvement (RPI) analysis is that the performance of setup time based algorithm, FJSRA increases continuously (continuous decrease in makespan) with increase in setup time. Hence, when setup time increases as proportion of processing time, the setup time based algorithm emerges as the better choice for solution. For smaller size problems, the greater portion of the RPI graph is in the negative range, while for larger size problems, the major portion of RPI graphs is in positive range. This also shows the dominant character of NEHRB heuristic for smaller size problems and the dominant character of FJSRA heuristic for larger size problems. The results of the statistical analysis not only validate the results of RPI analysis but also lead to the development of the setup time dominance matrix. This matrix is the consolidated statement of the statistical analysis carried for all the 96 groups of problems. The setup time dominance matrix originates from the logic that for any particular problem size, there exists a setup time level for which the setup time based algorithm, FJSRA becomes statistically superior. This frame work can be used for deciding the algorithm to be adopted for solving a particular size of problem.

References

1. Eren T.A. (2010), Bi-criteria m-machine flow shop scheduling with sequence-dependent setup times, Applied Mathematical Modeling 34(2): 284-293.
2. Gajpal Y., Rajendran C., Ziegler H. (2006), An ant colony algorithm for scheduling in flow shops with sequence-dependent setup times of jobs, International Journal of Advanced Manufacturing Technology 30(5-6): 416-424.
3. Gholami M., Zandieh M., Alem-Tabriz A. (2009), Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns, International Journal Advanced Manufacturing Technology 42(1-2): 189-201.
4. Gupta J.N.D. (1986), Flow shop Schedules with Sequence Dependent Setup Times, Journal of Operations Research Society Japan 29(3): 206-219.

5. Hwang H., Sun J.U. (1998), Production sequencing problem with re-entrant work flows and sequence dependent setup times, International Journal of Production Research 36(9): 2435-2450.
6. Mansouri S.A., Hendizadeh S.H., Salmasi N. (2009), Bi-criteria scheduling of a two-machine flow shop with sequence-dependent setup times, International Journal Advanced Manufacturing Technology, 40(11-12): 1216-1226.
7. Nawaz M., Enscore E.E., Ham I. (1983), A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega 11(1): 91-97
8. Naderi B., Ruiz R., Zandieh M. (2010), Algorithms for a realistic variant of flow shop scheduling, Computers & Operations Research 37(2): 236-246..
9. Naderi B., Zandieh M., Roshanaei V, (2009), Scheduling hybrid flow shops with sequence dependent setup times to minimize makespan and maximum tardiness, International Journal Advanced Manufacturing Technology 41(11-12): 1186-1198.
10. Rajendran C., Ziegler H. (2003), Scheduling to minimize the sum of weighted flow time and weighted tardiness of jobs in a flow shop with sequence-dependent setup times, European Journal of Operational Research 149(3): 513-522.
11. Rios-Mercado R.Z., Bard J.F. (1998), Heuristics for the flow line problem with setup costs, European Journal of Operational Research 110(1): 76-98.
12. Ruiz R., Maroto C., Alcatraz J. (2005), Solving the flow shop scheduling problem with sequence dependent setup times using advanced meta heuristic, European Journal of Operational Research 165(1): 34-54.
13. Salmasi N, Logendran R Skandari MR (2011) Makespan minimization of a flow shop sequence-dependent group scheduling problem, International Journal of Advanced Manufacturing Technology 56(5-8): 699-710.
14. Simons J.V. (1992), Heuristics in flow shop scheduling with sequence dependent setup times, Omega 20(2): 215-225.
15. Sun J.U., Hwang H. (2001). Scheduling problem in a two machine flow line with the N-step prior-job-dependent set-up times, International Journal of Systems Science 32(3): 375-385.
16. Wang H., Chou F., Wu F. (2011), A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan, International Journal Advanced Manufacturing Technology 53 (5-8): 761-776.