International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

Flow Shop Scheduling With Sequence Dependent Setup time

Abdul khayoom fawas
Head of Department in Automobile Engineering
Seethi Sahib Memorial Polytechnic College, Tirur

Date of Submission: 08-12-2022 Date of Acceptance: 16-12-2022

ABSTRACT
Sequencing and scheduling is a formof decision-making that plays a crucial roleinmanufacturing and service industries. In the current competitive environmenteffective sequencing and scheduling has become a necessity for survival in themarket-place. Companies have to meet shipping dates that have been committedto customers, as failure to do so may result in a significant loss of goodwill. Theyalso have to schedule activities in such a way as to use the resources availablein an efficient manner.

The present work deals with the problem of scheduling a flow shop operating in a sequence dependent set up time environment. The objective is to determine the sequence that minimizes the make span. The heuristic algorithm termed as Gap Optimization Technique (GOT) algorithm. This algorithm modify the existing sequence developed by SS-APST heuristic. This is based on the principle gap concept between machines and which is minimized by reducing the idle time of machines between the operations. The algorithms developed in the present work provides better make span results for sequence dependent setup time flow shop scheduling problem. The analysis revealed that the developed GOT algorithm has 5% decrease in make span value with respect to SS-APST algorithm.

I. INTRODUCTION

The present paper consider more realistic situation encountered in general flow shop scheduling by considering sequence dependent set up time (SDST) separately from processing time. This problem is known as SDST flow shop scheduling. These situations can be found in various production, service and information processing system.The constructive heuristic namely State Space Average Processing and Set up time (SS-APST) develops a scheduling sequence. This developed sequence are optimized in two stages for minimizing the make span. The first stage of optimization is done using improvement heuristic namely Gap Optimization Technique algorithm (GOT) and second stage of optimization is carried out using Variable Neighborhood Search (VNS) algorithm

II. LITERATURE REVIEW

Plenty of research work had been carried out in the field of flow shop scheduling problem

Vanchipura et al. **[1]**proposed how the concept of neighborhood search was used to develop two scheduling heuristics. These heuristics had been developed with the objective of minimizing the make span in a flow shop wherein the setup times are sequence dependent. These two algorithms NEHRB-VND and FJSRA-VND were developed by enhancing the capabilities of NEHRB and FJSRA, respectively by integrating with them the power of VND.

Framinan & Gonzalez [2] addressed the problem of scheduling jobs in a permutation flow shop when their processing times adopt a given distribution (stochastic flow shop scheduling problem) with the objective of minimization of make span.

Lia& Freiheitb [3] studied variation in sequential task processing times occurred in manufacturing systems. This type of disturbance challenges most scheduling methods since they cannot fundamentally change job sequences to adaptively control production performance as jobs enter the system because actual processing times, were not known in advance.

Afzalirad & Rezaeian [4]studied the case of an unrelated parallel machine scheduling problem with resource constrains, sequence-dependent setup times, different release dates, machine eligibility and precedence constraints.

INFERENCES DRAWN FROM LITERATURE REVIEW

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

The following inferences were drawn by reviewing the above mentioned literatures:

- Over the past five decades, extensive research has been done on n/m flow shop scheduling problem
- Sequence dependent setup time (SDST) version is relatively less explored by researchers due to complexity of SDST flow shop scheduling problem, which have been proved as NP complete problem.
- There are only less varieties of meta heuristics that can be applied to SDST flow shop scheduling problem.
- Due to complexity of SDST flow shop scheduling problem, development of constructive algorithms and meta heuristics poses a challenging task for researchers and practitioners
- For optimization of performance measures like make span, a standard design for constructive or meta heuristics algorithms are still under research

PROBLEM DEFINITION

The objective is to find a sequence for the processing of the jobs on the machines so that the total completion time or makes pan of schedule is minimized Therefore, an effort has made to minimize the time at which the last job in the sequence finishes the processing on its last machine, ie minimize the make span.

OBJECTIVES OF THE PRESENT WORK

Followings are the objectives of the present work:

a) To study the exiting methodology applied on general flow shop.

b) To develop a meta heuristic model for solving SDST flow shop environment.

III. METHODOLOGY

A flow shop scheduling problem involves a set of n jobs, tasks or items (1, 2 ….n) to be processed on a set of m machines or processor(1, 2…….m) in the same order.The objective is to find a sequence for the processing of the jobs on the machines so that the total completion time or makes pan of schedule is minimized. The desired objective has been achieved by following the methodology as shown in figure

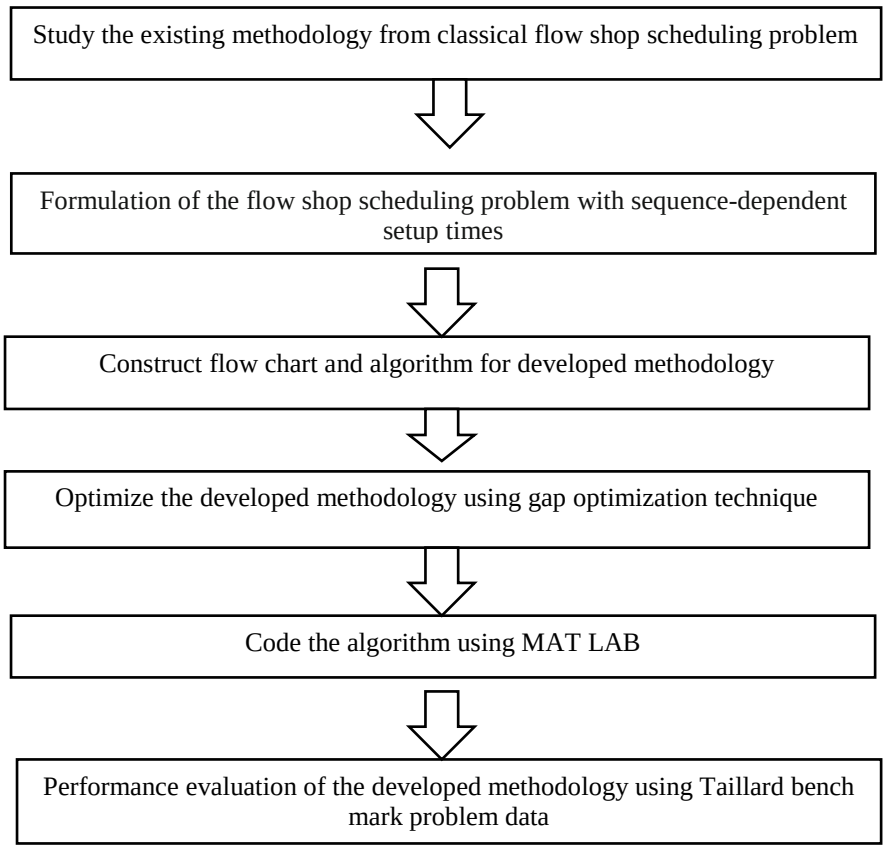


FIGURE : FLOW CHART OF METHODOLOGY

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

In order to solve the scheduling problem in a more realistic situation, a new method has developed which is named as state space average processing and setup time (SS-APST).The method deals with a more realistic environment encountered in general flow shop scheduling by considering sequence dependent setup time (SDST) separately from processing time.The solution generated by SS-APT heuristic can be optimized byGap Optimization Technique (GOT). This technique is based on the principle that minimize the gaps between successive operations in a solution generated by SS-APT so that total make span is optimized.

State Space Average Processing and Setup Time Method

The state space average processing and setup time (SS-APST) heuristic method is an adaptive control method used when there is variation in processing time and set up time. The objective is to find a sequence for the processing of the jobs on the machines so that the total completion time or make span of schedule in minimized. This method sequences jobs based on minimum of sum of current idle time and future idle time (ie minimum aggregate idle time).This method is based on the principle that will sequence the jobs one by one which is having minimum value of the sum of current idle times and future idle times. Present processing times and setup times of jobs are used to calculate current idle time whereas average processing times of unsequenced jobs are used to estimate future idle time.

Optimization of Developed Methodology Using Gap Optimization Technique

Gap Optimizing Technique (GOT)is an improvement heuristic for job scheduling in flow shop.This methodology is based on the principle that minimizes gap between successive operations in a solution generated by SS-APST heuristic. The sequence generated by the SS-APST heuristic is the input for the GOT heuristic algorithm. The final output is the optimized sequence with minimum make span. This method is explained as follow:

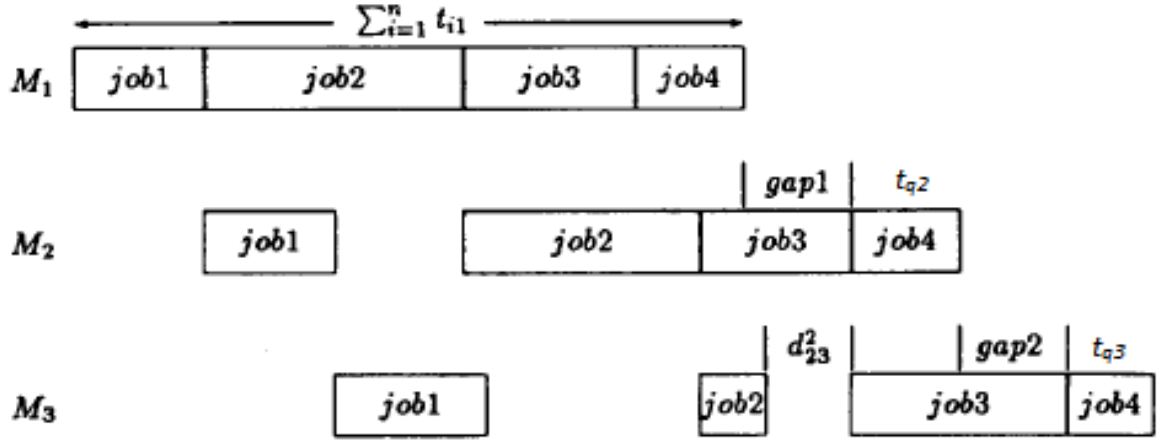


FIGURE : GANTT CHART FOR 4X3 SCHEDULING PROBLEM

Considering a three machine problem as shown in Figure 3.2.It is observed that, no idle time exists between jobs in machine -1. In the problem q is assigned to be the last job in the schedule, gap1 is defined as the time between the end of job q in M1 and start of job q in M2. Similarly, the time between the end of job q in M2 and start of job q in M3 is defined as gap2. Then the make span for n/3 problem is

Make span $= \sum_{i=1}^{n} ti1 + tq2 + tq3 + gap1 + gap2$ …. (3.1)

The first three elements in the equation are constant, and hence the total make span is optimized by minimizing the sum of gap1 and gap2. The new heuristic is designed to achieve minimum make span by minimizing these gaps. The time difference between successive jobs on adjacent machine can be written as

$d_{i,j}^{k} = t_{i,k+1} - t_{j,k}$ for $i, j = 1, 2, \ldots, n, k = 1, 2, \ldots, m-1$ and $i \neq j$ …. (3.2)

$t_{i.k+1=}$ finishing time of job i on $k+1^{th}$ machine
$t_{j.k=}$ finishing time of job j on k^{th} machine
m= number of machines

If the job i precedes job j in the schedule, the positive value of d^{k}_{ij} implies that job j needs to wait on machine $k+1^{th}$ at least d^{k}_{ij} until job i finishes. A negative value of d^{k}_{ij}implies there exist d^{k}_{ij} idle time between i and j in machine k+1.gap1 $=d^{1}_{34,}$ gap2$=d^{2}_{34}$ are positive gaps means job 4 will be

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

waiting and d^2_{23} is negative gapmeans machine idle time between job 2 and job 3 in machine 3
Following criteria are used to minimize above gaps (Gap optimization technique)

- Placing the pair of jobs with most negative gaps near the end of a schedule
- Placing the pair of jobs with the most positive gaps near the beginning of a schedule.

The negative value near the end of a schedule helps reduce the positive value accumulated earlier in the schedule, while the negative value near the beginning of schedule have high probability of being wasted (i e become idle).Positive gaps near the beginning help to compensate with negative gaps on later jobs.

3.1.4.1 Gap Optimizing Technique Heuristic Algorithms and Flow Chart
GOT heuristic algorithm optimize the sequence generated by the SS-APST heuristic by minimizing the gabs between jobs. GOT heuristic algorithm have eight important steps.
Step 0: Initialization.
Step 1: Constructing Gap matrix for all possible pair of jobs. Number of Gap matrix is equal to m-1, since there is no gap between jobs in first machine.
Step 2: Finding the discount factor (multiplication factor) for m-1 machines.
Step 3: Constructing revise gap matrix.
Step 4: Loading the initial sequence and find the makespan.
Step 5: Searching for largest value(X) and smallest value(Y) of revised gap matrix and identify the position in the sequence and named as U and V.
Step 6: Applying the condition and swap the jobs in the sequence.
Step 7: Finding the makespan of new sequence and if it is less than the original sequence, it becomes a new solution otherwise go to original sequence.
Step 8: Repeating the steps until the termination condition occur (b<=a+2).
The following steps is carried out for finding the revised gap matrix
Let $d^k_{i,j}$ gap matrix for all machine except machine 1 and m be the number of machines.
Before combining$d^k_{i,j}$, a factor k helps to discount the negative values.
factor $k = (\frac{1.0-0.1}{m-2})$ X $(m - k - 1) + (0.1)$ for $k = 1, 2, \dots, m-1$
Assign higher weights to small k and lower weights to large k.
For example
For k = 1, factor k = 1.0 and for k = m-1, factor k = 0.1
Intermediate machine use a linear interpolation of 1.0 and 0.1

Discount factor is determined as follows

$$\delta^k_{i,j} = \begin{cases} \text{factor k if } d^k_{i,j} < 0 \\ 1 \text{ otherwise} \end{cases}$$

for i, j = 1, 2, …., n and k = 1, 2, …M-1
Based on this factor, obtaining the overall revised gaps ie $d^R_{i,j}$as follows

$$d^R_{i,j} = \sum_{k=1}^{M-1} d^k_{i,j}\delta^k_{i,j} \text{ for } i, j = 1, 2, \dots, n.$$

Swapping procedure in initial sequence can be conducted as follows

1. Ifl (l=1,…..n) represent position in the incumbent solution and P_l represent the job in the position l, set a to 1, and b to n
2. Searching for largest value (called X) of $d^R p_a p_l$, where l is between a and b. Let U represent this l associated with X
3. Searching for the smallest value(called Y) of $d^R p_l p_b$, where l is between a and b. Let V represent this l associated with Y
4. If (X<0),(Y>0), and (IXI ≤ IYI), then going to step 7
5. If (X<0),(Y>0), and (IXI > IYI), then going to step 8
6. If (IXI >IYI), then going step 9. Otherwise going to step8
7. Let a=a+1, and swapping the jobs in the positions a and U
8. Let b = b -1, and swapping the job in positions b and V.
9. If the new schedule is better than the incumbent in terms of the performance measure, it becomes the new incumbent solution. Otherwise swap go to original incumbent solution.

The algorithm is initiated by loading the processing and setup time matrix. In the first step, constructing gap matrix for all possible pair of jobs. The next is to find the discount factor for all m-1 machines. Using this, constructing the revised gap matrix in the third step. Now loading the initial sequence generated by SS-APST heuristic and find its make span. In the fifth step, chosing the largest value and smallest value and identifying the position in the existing sequence. In the step six, applying the condition and swap the job. In the step seven, find the make span of new sequence and if it is less than original SS-APST, it is optimized sequence otherwise going to original sequence.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

Description of SS-APST MATLAB Programs Code

SS-APST heuristic is coded using MATLAB. The MATLAB program file SS-APSTSCRIP.m was attached in the Appendix 1. Initially load the processing and setup time matrices. The program sequences the job one by one to the final sequenced job set. Individual job sequencing is based on the principle that choosing a job which should have minimum aggregate idle time

IV. RESULTS AND DISCUSSION

The average make span of SS_APST and GOT algorithms were found and compared with existing three algorithms, NEHRB, SRA and FJSRA. The average make span of 10 Taillard bench mark problem instances of different algorithms for differentsizes such as 20x5, 20x10, 20x20, 50x5, 50x10, 50x20,100x5,100x10,100x20,200x10,200x20 and 500x20 are shown in the following tables. Relative performance index of GOT w.r.t NEHRB and FJSRA are also tabulated.

Table denotes the comparative analysis of average make span for the case of setup time as 5% of processing time. The results show the make span obtained using GOT method has minor variation when compared with other heuristic. GOT give better results in some problem instances when comparing the results with FJSRA algorithm. When comparing the results the NEHRB algorithm, the GOT method shows bad performance. GOT method perform better as compared with FJSRA for almost all cases of Taillard bench mark problem instances. The GOT method give better results when compared with FJSRA algorithm.for the problem sets of Taillard 20x5, 20x10, 20x20, 50x10 and 50x20.

TABLE : AVERAGE MAKE SPAN ANALYSIS FOR SETUP TIME AS 5% PROCESSING TIME OF TAILLARD BENCH MARK PROBLEM DATA

Sample number	Problem Size	Mean Make Span of algorithms					RPI of GOT w.r.t	
		NEHRB	SRA	FJSRA	SS-APST	GOT	NEHRB	FJSRA
1	SDST 20x 5	1294.8	1564.1	1367.6	1373.2	1350	-4.21	1.33
2	SDST 20x 10	1631.8	1951.4	1762.9	1756.7	1728	-5.87	1.96
3	SDST 20x 20	2366.6	2774.4	2530.7	2584.3	2514	-6.22	0.69
4	SDST 50x 5	2843.4	3202.2	2900.8	2930.5	2913	-2.47	-0.44
5	SDST 50x 10	3240.9	3965.5	3407.1	3415.6	3367	-3.87	1.17
6	SDST 50x 20	4068.8	4888.2	4283.9	4292.5	4262	-4.78	0.49
7	SDST 100x 5	5429.5	6146.3	5473.5	5606.9	5559	-2.39	-1.57
8	SDST 100x 10	5960.5	6947.1	6107.6	6228.5	6192	-3.89	-1.39
9	SDST 100x 20	6889.2	8061	7129.4	7331.1	7231	-4.96	-1.43

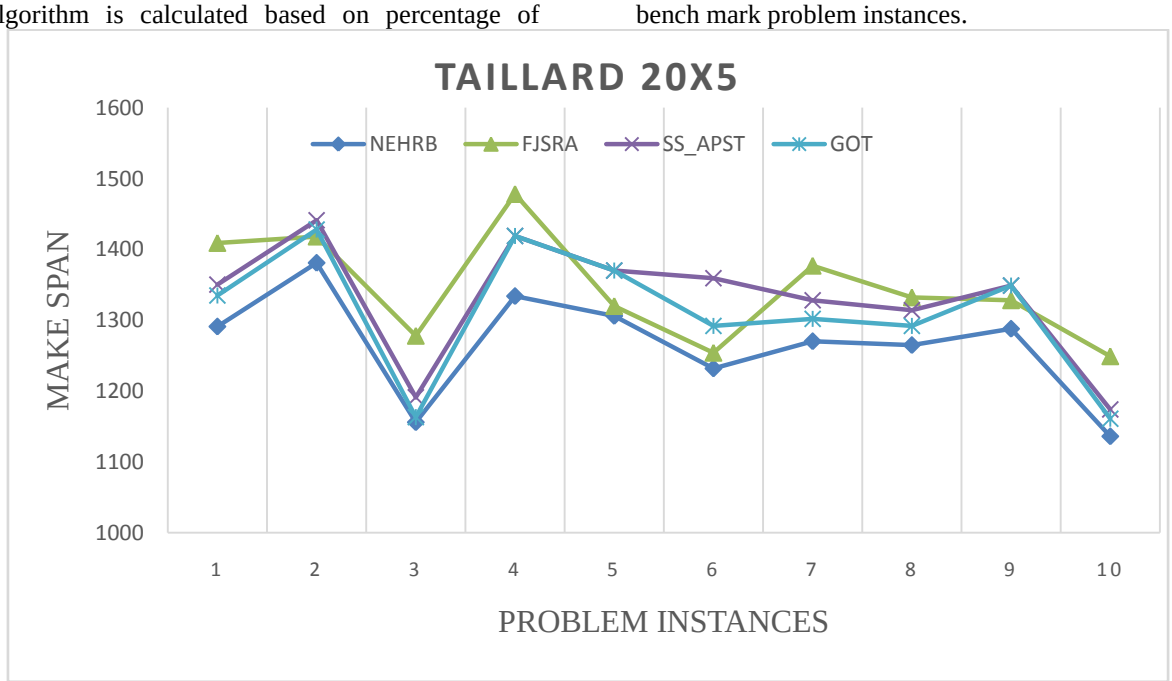
International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 12 Dec. 2022, pp: 332-338 www.ijaem.net ISSN: 2395-5252

10	SDST 200x 10	11149.8	12556	11311	11467.9	11410	-2.33	-0.87
11	SDST 200x 20	12183	14145. 5	12561.9	12694.8	12610	-3.51	-0.39
12	SDST 500 X 20	28046.5	31129	28409.3	28966.9	28785	-2.63	-1.32

From the result shown in Tables it can be generally observed that the value of make span obtained using GOT algorithm are less than that obtained using SS_APST algorithm. Ie results obtained using GOT algorithm are superior. It can also be seen that, results of GOT algorithm when compared with FJSRA algorithm give better result for the small size of benchmark problem data such as 20x5, 20x10, 20x20, 50x5, 50x10 at lower set up time. As the problem size increases, the results using FJSRA algorithm give little bit better performance. The result using NEHRB algorithm always shows better performance, but complexity of algorithm and flow shop environment become tedious to use these algorithm. Also adaptive control of SS_ APST algorithm always make it best for highly customized production environment.

The relative performance index of GOT algorithm is calculated based on percentage of improvement over two criteria, one is NEHRB and second is FJSRA. Both value are tabulated as RPI-NEHRB and RPI-FJSRA. A positive value of RPI indicate that proposed algorithm GOT perform better than NEHRB or FJSRA. Generally RPI of GOT w. r. t FJSRA shows better performance for small size of bench mark problem data at lower level of set up time. RPI of GOT w. r. t NEHRB shows varying value. Some cases such as 50x5 and 50x10, GOT is dominated at lower level of set up time.

The graphical analysis of SS-APST and GOT heuristic algorithms for Taillard 20x5 bench mark problem set are shown in Figure .The graphical analysis of standard NEHRB and FJSRA algorithms are also shown in figure. The graphs did not show any particular pattern of curve, but variation in every graphs are same for all Taillard bench mark problem instances.



GRAPHICAL ANALYSIS OF ALGORITHMS FOR TAILLARD 20X5 PROBLEM DATA

V. CONCLUSIONS

In the present work, two new heuristic algorithms are developed for flow shop scheduling with sequence dependent set up time. Taillard bench mark problems are used to develop SDST bench mark problem at nine different level of

sequence, with processing time matrix remain same as in Taillard bench mark problems. The algorithms SS_APST (State Space Average Processing and set up time) is a constructive heuristic and algorithms GOT (Gap Optimization Technique) is an improvement heuristic.The various conclusions drawn from analysis of results are as follows:

- The developed GOT algorithm has 5% decrease in make span value with respect to SS-APST algorithm. Hence the GOT algorithm has given the optimized value of make span.
- For the problem size SDST 20x5 and SDST 20x10, GOT algorithm achieve about 2% reduction in make span as compared with FJSRA algorithm in both 5 and10% setup time version.
- The analysis generally concluded that at low percentage of setup time, GOT algorithm perform better and variation exist within the range of -5% to +5% and hence the developed algorithm can be taken as standard algorithm for general flow shop scheduling problem

The analysis of present work has endless scope for future study. Following are the area in which future work can be done.

- Opportunities to include sensitivity of SS_APST heuristics to variation in processing time will be examined to further improvement its consistency and performance.
- Both algorithms can include the set up variation with respect to the job in the sequence.
- Algorithm can be enhanced by including the various fluctuation in the production environment.

REFERENCES AND BIBLIOGRAPHY

[1] Vanchipura R., Sridharan, R.& Babu, S. A., “Improvement of constructive heuristics using variable neighborhood descent for scheduling a flaw shop with sequence dependent setup times”, Journal of Manufacturing systems, vol.33, 2014, pp.65-75.

[2] Framinan, J.M.& Gonzalez, P.P.,“One heuristic for stochastic flow shop scheduling problem”, European Journal of Operational Research, vol. 246, 2015, pp. 413–420.

[3] Wei, L.& Freihet, T.I., “ An effective heuristic for adaptive control of job sequences subject to variation in processing times”, International Journal of Production Research, vol.28, 2015, pp.1-17.

[4] Afzalirad, M.& Rezaeian, J., “Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions”, Computers and Industrial Engineering ,vol. 45, 2016, pp.1-47.