

**SOLVED m- MACHINES, n- JOB FLOW SHOP SEQUENCING PROBLEM USING A
HEURISTIC ALGORITHM**

ANUJ GANGWAR

(M.Tech Scholar) Department of Mechanical Engineering
Shri Ram Murti Smarak College of Engg. & Technology
Bareilly, India
anujgangwar000@gmail.com

ASHWANI KUMAR

(Asst. Professor) Department of Mechanical Engineering
Shri Ram Murti Smarak College of Engg. & Technology
Bareilly, India
ashipe90@gmail.com

Abstract

Flow shop sequencing defines that jobs will be passed in a sequence from first to the last machine. Many heuristic algorithms introduce that the jobs with greater completion time or processing time should be given higher preference than the jobs with lower total process time. Hence, in this paper according to this approach, a simple algorithm is presented which give the best sequence in comparison to other existing heuristic. The results of proposed algorithm are then compared with the results of 15 other algorithms in a separate study by Park, who displays that the proposed algorithm's outcomes especially well on the broad flow shop sequencing problems.

I. INTRODUCTION

Scheduling is the allocation of resources over time to perform a collection of tasks and it is a decision-making process that exists in the flexible manufacturing system that increases productivity. Main resources of the system are machines, tools, material handling equipment and storage and retrieval system[3].

In the flow shop sequencing problem rank, order etc. over a set of jobs (task, items) are processed from one machine to another, hence we say that in the flow shop sequencing problem related to determining the relative position of jobs (i) to all other jobs. These problems have occurred in many different environments. They exist whenever there

is a choice as to the order or rank in which a number of tasks (jobs) can be performed. So that problem exists jobs in manufacturing plant, computing center and aircraft waiting for landing clearance. In them-machines, n-jobs flow shop sequencing to minimize makespan time, Johnson's proposed several algorithms. They also give the solution of 2-machine and specially 3-machine flow shop problems. In all algorithms researcher taken more restrictive drawbacks, though far from the real situation greatly simplify the problem. Ali Allahverdi [4] also gave the complete list of drawbacks. These are given below-

A. Assumption regarding machines:

1. No machine may process more than one job at any given time and each job, once started, must be processed to completion.
2. A known, finite time is required to perform each operation and the time intervals for processing are independent of the order in which the operation are performed.

B. Assumption regarding jobs:

1. All jobs are known and are completely organized for processing before the period under consideration begins.
2. All jobs are considered equal in importance that there are no due dates.
3. Jobs are processed by the machines as soon as possible and in a common order.

C. Other:

1. The time required to transport job between machines may be considered negligible or as a part of the processing time on the preceding machine.
2. In-process inventory is allowable.

In this paper, researcher search the solution for finding an optimal and near-optimal sequence has yielded both exact and approximate solution technique. In most cases, the computational time and the memory of required to keep track of the calculation are prohibitive even for small problems but the exact techniques solve the problem in principle. In other words, we say that the heuristics approach do not give the optimal solution of the

problem its mean mostly part of the algorithm are an efficient and economical way to finding the best solution of the problem.

Palmer proposed a slope order index to sequence n jobs on m machines based on the processing time of jobs on different machines then another algorithm in which defined slope index in a slightly different manner [1]. Cambell, Dudek, and Smith (CDS) [5] proposed a procedure in which heuristic generalization of Johnson's algorithms. In this procedure, CDS generates a set of m-1 artificial two machine problem from the original m machine problem. In which all are the problems is solved by Johnson's algorithms.

Hence in the flow shop sequencing problem to find a near optimal solution researcher give many algorithms. In the past few years, many researchers find the well-known heuristics have statistically evaluated the performance. According to the Nawaz [6] in the procedure of multi ranking the CDS heuristics shown superior to the other heuristics.

A new curtailed enumeration algorithm is proposed for the flow shop sequencing problem in the following section.

II. THE PROPOSED ALGORITHM

The flow shop sequencing problem can be stated as follows:

Given in the same order n jobs to be processed on m machine and the processing time of jobs (i) on machine (j) being $t(i,j)$ [$i = 1,2,3,-----n; j = 1,2,---$

----- m] so to minimize the makespan or total completion time we find out the sequence of jobs.

In the flow shop sequencing problem also presented above it is a combinatorial search problem with $n!$ The sequence in this for identified the minimum total flow time or completion time then! Sequence enumerated.

In the proposed algorithm we will take an assumption which is based on the job with more total processing time on all the machine given higher priority than a job with less total Process time. Hence the overview of the proposed algorithm can be stated as follows.

In this algorithm from n jobs, we selected two jobs with the highest total processing time and then in which we found the best partial sequence by using an exhaustive search that considered in two possible partial schedules [2]. Then we fixed the relative position of these jobs with respect to each other jobs and in remaining steps of the algorithm. Next, we find out the third job with the highest total processing time be selected and then we find out the three partial sequences.

In all these sequences we tested job placed at the beginning, middle and end of the partial sequence found in the first step. Hence the best partial sequence in all of the sequences will be fixed and the relative position of the three jobs for the remaining steps, so this process is repeated again and again until all the jobs are fixed and find a complete sequence. In this algorithm we also find out the number of enumeration according to below formula:

$$[n(n+1)/2]-1$$

In which n is an enumeration of the complete sequence which is used in this algorithm and rest are the partial sequence.

So for the curtailed-enumeration algorithm, we followed the step by step procedure that is given below-

Step1. Calculate every job processing time (i)

$$T_i = \sum_{j=1}^m t(i,j)$$

Where $t(i,j)$ is the processing time of job ion machine j .

Step2. We arrange the job in descending order of T_i .

Step3. In step 2 we pick the two jobs from first and second position of the list and calculate makespan for the two possible sequences to find out best sequence for these two jobs and hence the relative position of these two jobs do not change with respect to each other in the remaining steps of the algorithm and set $i = 3$

Step4. In step 2 a list generated in this we pick the job in its position and then we placed i th position in all possible sequence for finding the best sequence. We found in the previous step all jobs are already assigned so do not change relative position to each job hence we see the number of enumeration at this step that is equal to i .

Step5. If $i=j$ then STOP, otherwise set $i=i+1$ and go to step 4.

III. NUMERICAL ILLUSTRATION

To illustrate the procedure outlined above hence we taken 4 jobs and 5 machine flow shop problem as shown in (Ref Table- 1).

Operation Time Matrix

Step1. $T1 = 6+10+9+11+2 = 38$

$T2 = 10+4+11+2+9 = 36$

$T3 = 10+5+6+9+7 = 37$

$T4 = 5+9+9+8+3 = 34$

Step2. Hence we arrange the jobs according to processing time, and then the sequence is 1-3-2-4

Step3. In this we pick the job 1 and job 3 to find out the optimal partial sequence for these two jobs that is given below table- (Ref Table- 2 & 3)

Hence according to Table 2 and Table 3 it is clear that the sequence 3-1 is the best with a makespan 48. So in the next step, the relative position of job 1 and job 3 should always be fixed 3-1 and set $i = 3$

Step4. Now in this step, we take the job in the third position of the list of step 2 and then placing job 2 in all three possible positions to find the optimal sequence. In this, we find the makespan for all three possible sequences that are given in below table- (Ref Table- 4, 5 & 6)

Hence we see in table 4, 5 and 6 the minimum makespan find in table 4 its means 3-1-2 is the best sequence with makespan 57.

Step5. We see according to this solution the number of enumeration is not equal to the number of jobs is means:

$$i \neq n$$

Hence, $i = i+1 = 3+1 = 4$ then we go to step4.

Step6. In this, we pick the job in the fourth position of the list of step 2 and then placing 4 jobs in four possible positions of the partial sequence 3-1-2 to find out the optimal sequence. Then the complete sequence of all four jobs are given in the below table- (Ref Table- 7, 8, 9, & 10)

Now we see in table 7, 8, 9 and 10 the minimum makespan of 62 in table 10. Hence we say that the sequence 4-3-1-2 is the best sequence to find out the optimal solution.

Step6. $i = n$. hence stop

So this problem was solved for the optimal makespan by calculating all $n! = 24$ sequences and find out the 4-3-1-2 is the best sequence to give the optimal solution. Hence we see the proposed

algorithm has made 9 enumerations and in which 5 were the partial sequence and 4 enumerations were a complete sequence and we noted that if the number of enumeration will increase if in partial sequence find ties sequence and each tied sequence is examined.

Hence we observed that the proposed algorithm is best executed via a computer program for large flow shop problem. The FORTRAN listing for such a program is given Nawaz [6].

IV. EFFECTIVENESS OF THE PROPOSED ALGORITHM

The researcher tries to demonstrate the effectiveness of the proposed algorithm was to solve a large number of flow shop issues with the projected algorithm and compare the results with those get it by applying the CDS algorithm. Setiাপutra as previously discussed so these results are compared CDS algorithm. Hence the set level of 4 to 25 totally 2764 flow shop problems were solved with the number of machines and jobs. The processing time for the jobs was randomly generated from a uniform distribution with vary one to ninety-nine. The results are given in table 11, which is given below. Hence they have seen the proposed algorithm given better results or perform well as others.

Since this primary try at analysis Park has completed a most comprehensive study of flow shop algorithms. In this study, they evaluated 16 algorithms. And they studied the sensitivity according to problem size and computational potency. Hence in this paper study taken some statements and given the result [6]:

1. It is clear that NEH (Nawaz-Ensure-Ham) is the least biased and the best-operated of the heuristics tested on the small static flow shop problems (3 to 9 jobs combined with 4 to 20 machines) with respect to makespan and that CDS is the next best algorithm to finding the solution.

2. As in the case of the small size problems, it is clear from the test results that for minimizing makespan in large static problems (15 to 30 jobs combine with 4 to 20 machines) NEH is the least biased and most effective in this case.
3. The results of computational time (for large problems) indicate NEH is less attractive.

Hence we see the statement 1 and 2 show a clear dominance of the proposed algorithm with relation to statement 3 it ought to be noted that the typical computational time for the larger issues was only 0.3064sec. This can be massive when compared to Gupta and Boney & Gundry [1] algorithm which is given 0.0082 sec. and 0.094 sec. hence we see the 0.3064sec definitely does not represent excessive computer time. The CDS algorithm on an averaged 0.0885sec.

V. CONCLUSION AND FUTURE SCOPE

A new curtailed-enumeration algorithmic program for resolution the m-machine, n-job flow-shop problem has been represented during this paper. The algorithmic program is straightforward to know and apply. Like most heuristic algorithms accustomed to solving the flow-shop downside, the utilization of the computer is suggested once resolution issues with large n.

In this, we have seen the proposed algorithm has been shown superior to the CDS algorithm and other established algorithm for the range of jobs (3 to 30) and machines (4 to 25). In this evaluation carried out independently [6]. Even though no attempt was made at evaluating the proposed algorithm with a large number of machines and jobs. Hence they felt

that it would continue to perform well. In this, if the number of machines is more than the number of jobs then we assured that the CDS algorithm would outperform the proposed algorithm since the former effectiveness is dependent on the number of jobs.

In the flow shop, the completion time of the proposed algorithm is dependent on the number of jobs. In this, the minimum number of complete and partial sequence is considered by the algorithm is-

$$[n(n+1)/2]-1$$

Hence the average computer solution time for n ranging between 15 and 30 is 0.3 sec. in the CDS algorithm this time is 0.09sec. It means the proposed algorithm computational time is much greater than the CDS algorithm time.

Hence, we conclude that the proposed algorithm performs exceptionally well against established algorithm in solving the flow shop sequencing problem.

IV. REFERENCES

1. Nailwal k, Gupta D, and Sharma S, “*Minimizing makespan in flow shop under no-wait restriction on jobs,*” International Conference on Electrical, electronics and Optimization Techniques (ICEEOT), 2016.
2. Abedinnia H. and Glock CH, “*New simple constructive heuristic algorithm for minimizing total flow time in the permutation flow shop scheduling problem,*” Computer and Operation Research Elsevier, Vol. 74, pp. 165-174, 2016.
3. D. Laha and U.K. Chakraborty, “*A constructive heuristic for minimizing makespan in no wait flow shop scheduling,*” Int. J. Adv. Manuf. Tech. Vol. 41 (1-2), pp. 71-83, 2002.
4. Hamada G. Shen W. and Wang C, “*Heuristic scheduling algorithm for flexible manufacturing system with partial overlapping machine capabilities,*” Proceeding of the IEEE Int. Conference on Mechatronics and Automation July, 2005.
5. Allahverdi A. and Aldowaisan, “*New heuristic to minimize total completion time in m machine flow shop,*” Int. J. Production Economics Vol. 77, pp. 71-83, 2002.
6. Nawaz M, “*A heuristic algorithm for the m-machine n-job flow shop sequencing problem,*” OMEGA Int. J. of Mgmt. Sci. Vol. 11, No. 1, PP. 91-95, 1983.
7. Campbell HG, Dudek RA and Smith ML, “*A heuristic algorithm for the n job m machine sequencing problem,*” Mgmt. Sci. 16, (16), 1970.

LIST OF TABLES:

Table No.1

m/ c jobs	1 (sec)	2 (sec)	3 (sec)	4 (sec)	5 (sec)
1	6	10	9	11	2
2	10	4	11	2	9
3	10	5	6	9	7
4	5	9	9	8	3

Table2- Make Span for Partial Sequence 1-3

m/ c jobs	1	2	3	4	5
1	6/6	10/1 6	9/2 5	11/36	2/3 8
3	10/ 16	5/21	6/3 1	9/45	7/5 2

Table3- Make Span for Partial Sequence 3-1

m/c jobs	1	2	3	4	5
3	10 /1 0	5/1 5	6/2 1	9/30	7/3 7
1	6/ 16	5/2 6	9/3 5	11/46	2/4 8

Table4- Make Span for Partial Sequence 3-1-2

m/ c	1	2	3	4	5

jobs					
3	10/10	5/15	6/21	9/3 0	7/3 7
1	6/16	10/2 6	9/35	11/ 46	2/4 8
2	10/26	4/30	11/4 6	2/4 8	9/5 7

Table5- Make Span for Partial Sequence 3-2-1

m/c \ jobs	1	2	3	4	5
3	10/10	5/15	6/21	9/30	7/ 37
2	10/20	4/24	11/35	2/37	9/ 46
1	6/26	10/36	9/45	11/56	2/ 58

Table 6- Make Span for Partial Sequence 2-3-1

m/c \ jobs	1	2	3	4	5
2	10/10	4/14	11/25	2/27	9/36
3	10/20	5/26	6/31	9/40	7/47
1	6/26	10/36	9/45	11/5 6	2/58

Table7- Make Span for Sequence 3-1-2-4

m/c job	1 (sec)	2 (sec)	3 (sec)	4 (sec)	5 (sec)
3	10/10	5/15	6/21	9/30	7/37
1	6/16	10/26	9/35	11/46	2/48
2	10/26	4/30	11/46	2/48	9/57
4	5/31	9/40	9/55	8/63	3/66

Table8- Make Span for Sequence 3-1-4-2

m/ c jobs	1 (sec)	2 (sec)	3 (sec)	4 (sec)	5 (sec)
3	10/10	5/15	6/21	9/30	7/37
1	6/16	10/26	9/35	11/46	2/48
4	5/21	9/35	9/44	8/54	3/57
2	10/31	4/39	11/55	2/57	9/63

Table9- Make Span for Sequence3-4-1-2

m/c jobs	1 (sec)	2 (sec)	3 (sec)	4 (sec)	5 (sec)
3	10/10	5/15	6/21	9/30	7/37
4	5/15	9/24	9/33	8/41	3/44
1	6/21	10/34	9/43	11/54	2/56
2	10/31	4/38	11/54	2/56	9/65

Table10- Make Span for Sequence 4-3-1-2

m/c jobs	1 (sec)	2 (sec)	3 (sec)	4 (sec)	5 (se c)
4	5/5	9/14	9/23	8/31	3/34
3	10/1 5	5/20	6/29	9/40	7/47
1	6/21	10/31	9/40	11/51	2/53
2	10/3 1	4/35	11/51	2/53	9/62

Table11- Comparison of Proposed Algorithm with CDS Algorithm

Number of Jobs	Number of Machines	Number of Problems	Number of times solution better by the proposed algorithm	Number of times solution equal by proposed and CDS algorithm	Number of times solution better by CDS algorithm
4	4	20 0	41	153	6
4	5	20 0	43	152	5
4	6	20 0	53	142	5
4	20	10 0	14	67	1 9
5	6	20 0	77	105	1 8
5	8	20 0	66	115	1 9
5	10	20 0	72	98	3 0

ELK ASIA PACIFIC JOURNAL OF MECHANICAL ENGINEERING RESEARCH

ISSN 2349-9368 (Online); EAPJMER/issn. 2349-9368/2016; Volume 4 Issue 2 (2018)

5	20	10 0	38	43	1 9
6	8	20 0	103	67	3 0
6	10	20 0	95	75	3 0
6	12	20 0	95	70	3 5
6	20	10 0	55	17	2 8
10	10	20 0	185	10	5
10	15	20 0	175	9	1 6
10	20	20 0	176	10	1 4
15	15	25	25	0	0
20	20	25	23	0	2
25	25	14	14	0	0

