Vol. XVI & Issue No. 03 March - 2023
INDUSTRIAL ENGINEERING JOURNAL

A HEURISTIC APPROACH FOR SOLVING PERMUTATION FSSP WITH TCT OBJECTIVE

R Durgadevi
Dr. S Jayakumar
K Moorthi

Abstract

Scheduling problem as its origin in manufacturing industries. Particularly flow shop scheduling problem occurs in the timetable algorithm, operatory theory and combinatorial optimization problem to minimize the TCT (Total Completion Time) is a herculean task to obtain a near-optimal solution. Researchers are interested in solving these types of problems with the objective are TCT is quite fashionable. This paper examines the permutation FSSP for TCT. Also, Permutation FSSP has wide field applications in operation theory, combinatorial algorithms, and operations management. In this examination, we have fostered an algorithm that beats the algorithm accessible in the literature. However, the issue of three or more machines is recognized to be NP, resulting in near-optimal solutions, the method is simple and elegant to the given problem.

Keywords: Flow shop scheduling, Johnson's technique, Gupta's method, heuristic, TCT.

1. INTRODUCTION

Scheduling and Sequencing assume a critical part in the genuine world circumstances, creation, and manufacturing ventures for expanding the efficiency, working on the nature of items, satisfying the requests of the market in time, and limiting the flow time, the inactive time, and rental expense of the machines. Scheduling has numerous applications, in actuality, useful circumstances. Scheduling Problem created in numerous spaces, for example, manufacturing enterprises, transportation, medical care, clinics, PC programming, creation arranging, roller-bearing businesses, material ventures, building developments, in protection, etc. Sequencing just alludes to the assurance of request where the jobs are to be handled on different machines. Scheduling implies the time table that consolidates the starting time and completion times of jobs of machines, etc. Resources are commonly called Machines and tasks are called jobs or assignments. The climate of the scheduling problem is known as the shop. There are various kinds of shops utilized in scheduling problems like the flow shop, open shop, job shop, mixed shop, and so forth. Sequencing in the flow shop scheduling problems is too convoluted when the number of jobs and machines increments significantly Johnson (1954) was the first introduce Flow Shop Scheduling problem. Later Gupta (1971) developed on heuristic algorithm for FSSP while sequencing. For the two machine problem with makespan objective Meganathan et al (2016) developed an algorithm which is an excellent one. For the permutation flow shop scheduling problem Jayakumar (2019) work well. Unadulterated sequencing is a generally excellent region for research in solving FSSP to minimize make span. Since every single activity is in accessible rivalry with different capacities for restricted resources of time and amount, the job scheduling problem is neither basic nor simple. Sequencing is for each machine in the shop, one needs to assemble the request where in the jobs sitting tight for administration before that proper machine must be processed Sequencing and scheduling was introduced by Baker (1974). Scheduling the open shop problem is a tedious task. Jayakumar(2000) solved one such problem with the objective of minimizing the makespan and resource idleness. Later Jayakumar and Meganathan(2014) solved the OSSP with the objective of minimizing the makespan and release dates. The scheduling problem is a portion of resources over the long run to play out a grouping of assignments. This very broad meaning of the term does convey two particular implications. The first scheduling problem is the dynamic capacity. Second scheduling problem is the assemblage of the hypothesis of the assortment of rules, models, strategy, and an obvious result that gives experiences into the scheduling capacities. Scheduling is portrayed as relegating each course of each job a beginning time and finishing time size of machines inside the priority relations. Sequencing is a mechanism for organizing tasks in a certain order. Various sequencing types are gone on in enterprises like earliest in, earliest out premise, need premise, job size reason and handling time premise, etc. We shall accomplish assorted handling time by handling time premise sequencing for distinct sequences. The sequence is altered to provide for the least handling time. By scheduling, we assign a certain amount of time to do a specific activity. The main goal of scheduling is to arrive at a position where we will have the least amount of handling time. In this paper, we introduce the Total Completion Time (TCT), which is the sum of all the jobs on the last machine to complete. We aim to minimize TCT compared to Gupta's heuristic algorithm given below.

2. FLOW SHOP SCHEDULING

It's a combinatorial optimization challenge to schedule a flow shop. In a general FSS problem, we are provided n jobs J_1, J_2, ..., J_n each with a different processing time, that must be scheduled on m machines with shifting preparation power, while attempting to lessen the TCT –the absolute length of the schedule (i.e., the point where all of the jobs involving handling have been completed). In the particular variation known as flow shop planning, each job consists of precise m operations. The job's i^{th} operation should be completed on the i^{th} machine. At any one moment, no machine can accomplish more than one task. For every operation of each job, execution time is determined. The optimal sequence (order) indicates the base time wherein jobs, gear, individuals, materials, offices, and any remaining assets are organized to help the creation schedules to give low expenses and high uses. If n jobs are to be performed, and the genuine (or expected) time needed by the jobs on every one of the machine are given, then, the basic sequencing problem at that point is to determine a sequence out of $(n!)^m$ possible sequences that reduce the overall elapsed time between first machine's task initiation and final machine's end job. Specifically, in case there are n=5 jobs to be executed and machines m=5 is to be utilized, then, at that point, the total possible sequences can be $(5!)^5$ = 24,883,200,000. Hypothetically, it could be possible to track down the optimal sequence yet this may need a great deal of computational time. Consequently, one ought to embrace the sequencing strategy. To track down the sequence, firstly we need to compute the total elapsed time for every possible sequence. Scheduling is the way toward organizing, controlling, and optimizing work and responsibility in a manufacturing interaction. Flow shop scheduling where there is a severe request, everything being equal, to be performed on all jobs. Despite the fact that flow shop scheduling has its starting point in manufacturing enterprises. It has been generally utilized in the assistance area Jayakumar (2019) solved the permutation FSSP with makespan objective. Again Vasudevan et al (2020) solved the generalized n job m machine scheduling Flow shop problem with the same objective. Later Jayasankari (2021) solved the same problem with an effective algorithm. Vasudevan and Jayakumar (2020) solved the FSSP with the objective of minimizing the makespan and compared with the existing algorithms available in the literature. In this article, we have been focusing on finding the total completion time (TCT). The choice of request in which the jobs will be prepared (sequenced) is called the sequencing issue. This is a request to make powerful use of accessible resources. This sort of issue is an endless supply of resources either man-made or machine-made. The issue of discovering a sequence among the $(n!)^m$ sequences for handling the jobs which give the least TCT. Flow shop scheduling concludes an optimal sequence of n jobs to be taken care of on m machines in this single request, for instance, each job should be dealt with on machines1,2,..., m in the identical request. The FSSP is a manufacturing problem where a bunch of *n* jobs should be taken care of on m machines with indistinct flow plans. We have a permutation flow shop sequencing creation climate when the processing operation sequence on all machines is quite similar. A flow shop contains m machines in series and n different jobs accessible for processing at zero time. As a result, each machine may only do one job at a time. In this specialized order, each job is persistently handled on m accessible machines. This makes it unique to a common flow shop problem. The processing time including the arrangement time of the work done on the machine relies upon how much-restricted resources are distributed for these jobs.

3. PERMUTATION FLOW SHOP SCHEDULING (PFSP)

For instance, the order in which tasks are processed on every machine is identical for all machines, making permutation flow shop scheduling an exceptional case of FSP. The PFSP issue is a manufacturing problem for determining the optimum sequence of tasks that machines should handle to minimize a specified objective limit. This situation is common in manufacturing workplaces, where material handling devices are used to transmit tasks (parts) from one machine to the next, with no passing permitted. The problem is shown to be NP-complete, with the total number of feasible sequence (schedules) being for jobs. This difficulty is ended up being emphatically NP-complete, and the total number of possible sequences (schedules) of it for jobs. Make span and complete flow time are significant execution parameters that contribute to faster job turnaround and minimize in-process stock. The issue of stage flow shop planning is regarded as NP-hard. In the presence of mind, a planning problem is NP-hard if a segment (or a comparable problem) may be deduced to this problem with a polynomial-time calculation, and there is a computation with pseudo-polynomial time unpredictability that solves the booking problem. That is, as the size of the problem grows, the time needed to resolve calculation increases exponentially.

4. ASSUMPTIONS

The investigation has been finished by considering a few suspicions which are:

1. Each job must be handled on every machine in request (j=1,2,….,m)
2. Each machine measures just each job in turn
3. Each job is handled on each machine in turn
4. The times assigned to each job's setup are noted and used in the preparation process.
5. It is not permissible to pre-empt.
6. Machines never separate and are accessible all through the scheduling period.
7. It is assumed that the main machine will be ready for whatever task is assigned to it initially.
8. Some machines may be inactive (idle)

5. GUPTA'S HEURISTIC ALGORITHM

Gupta recommended a heuristic which was like Palmer's heuristic, in the year 1971.He described the slope index in a substitute manner by thinking about a few engaging real factors regarding optimally of Johnson's norm for the three machine issues. Johnson's calculation is essentially utilized for just two machines, yet the possibility of Gupta calculation is pertinent for multiple machines. This calculation expresses an m machine, a bunch of n free jobs with a progression of tasks that should be executed in a similar request on each machine. Gupta proposes heuristic calculations that can be applied to huge size issues in any event, for hand calculation.

Strategy:

Gupta's Heuristic Input:

i - job list, m - machine;

Yield: plan

S; start

For i= 1to n

for k=1 to m-1 if $t_{i1}<t_{im}$

then at that point e_i=1; else e_i= -1;

Step1: To find $S_i=e_i/\min\{t_{ik}+t_{i,k+1}\}$;

Step2: By indexing the tasks in non-increasing order of Si, the permutation schedule is created i.e., $S_{i1}\geq S_{i2}\geq$ .... $\geq S_{in}$

Step3: Schedule S represents the output optimal sequence.

Finish;

6. PROPOSED NEW MODIFIED ALGORITHM FOR PERMUTATION FSSP

The proposed newly modified algorithm is described here to provide a solution to 'm' machine n-jobs flow shop scheduling TCT problem for minimum TCT.

Jaya Durgadevi algorithm to solve n-job m-machine problem where n>m

Step 1: Check to see whether the job number exceeds the number of machines, then go to step 2 where m and n are greater than two.

Step2: Calculate the Sum of possible of all m-1 machines and it is denoted by g_i

Step 3: Find

$$si = \frac{1}{mingi}$$

Step 4: The permutation schedule is made by indexing the tasks in a non-increasing sequence of s_i i.e., $s_{i1}\geq s_{i2}\geq$......... $\geq s_{in}$.

Step5: Schedule S represents the output optimal sequence.

Numerical Example: Consider 6jobs, 5machine FSS problem with Consummation time about all jobs functioning all machines as shown inTable2.

Table1: Numerical illustration

Job/ Machine	M1	M2	M3	M4	M5
j1	2	5	4	2	5
j2	6	3	7	3	6
j3	5	4	6	5	4
j4	3	6	3	2	3
j5	4	2	5	4	4
j6	2	5	8	7	5

TCT calculation using Gupta's Algorithm

We get the sequence is j1,j6,j3,j2,j5,j4

TCT of Gupta's algorithm=223

Solution using Jaya Durgadevi algorithm

Step1: Assume no. of jobs by n and no. of machines by m where (n>m).

Step 2: We find Sum of possible of all m-1 machines are g_1= (16,13,14,16,13);
g_2= (19,22,18,22,19);
g_3= (19,20,18,19,20);
g_4= (14,11,14,15,14);
g_5= (15,17,14,15,15);
g_6= (25,22,19,20,22).

Step3:

$$s_1 = \frac{1}{ming_1} = \frac{1}{13} = 0.077;$$
$$s_2 = \frac{1}{ming_2} = \frac{1}{18} = 0.056;$$
$$s_3 = \frac{1}{ming_3} = \frac{1}{18} = 0.056$$
$$s_4 = \frac{1}{ming_4} = \frac{1}{11} = 0.091$$
$$s_5 = \frac{1}{ming_5} = \frac{1}{14} = 0.071$$
$$s_6 = \frac{1}{ming_6} = \frac{1}{19} = 0.053$$

Step4: Non-increasing order of s_i as $s_4\geq s_1\geq s_5\geq s_3\geq s_2\geq s_6$

Step5: we get the optimal sequence is obtained as schedule {j4,j1,j5,j3,j2,j6}

Processing time about all jobs functioning all machines for the sequence {j4,j1,j5,j3,j2,j6} is given in the following Table.

Table 2: TCT calculation using our new modified algorithm for flow shop scheduling problem

Job/Machine	M1	M2	M3	M4	M5
j4	0-3	3-9	9-12	12-14	14-17
j1	3-5	9-14	14-18	18-20	20-25
j5	5-9	14-16	18-23	23-27	27-31
j3	9-14	16-20	23-29	29-34	34-38
j2	14-20	20-23	29-36	36-39	39-45
j6	20-22	23-28	36-44	44-51	51-56

We get TCT = 212

The best sequence of Jaya Durgadevi algorithm based on TCT is {j4,j1,j5,j3,j2,j6}

7. RESULT ANALYSIS

In the result analysis we compare the problems for the number of machines less than number of jobs. Here we compare our result with the Gupta's algorithm to our algorithms. It has been found that our algorithm performs better than Gupta's algorithms found in the literature. The Table 3 below gives the comparison of the algorithms found in the literature. A sample of 12 such problems chosen to show the effectiveness of the algorithm and the problems are given in appendix.

Table: 3

Instances	nxm n- jobs, m-machines	Total Completion time (TCT)	
		Gupta's Algorithm	Our Algorithm
1	4x3	49	45
2	4x3	92	87
3	4x3	91	71
4	5x3	97	96
5	5x3	96	92
6	5x4	113	105
7	5x4	127	116
8	6x4	130	129
9	6x4	160	159
10	6x5	310	271
11	6x5	223	212
12	7x3	178	177

8. CONCLUSION

This paper proposed another flow shop scheduling problem for two or more machines. This algorithm gives much better results than Gupta's heuristic. It gives less TCT like better partial TCT. In addition, the proposed algorithm provides several choices for the best outcome. With TCT, rather than Gupta's heuristic, we obtain a sequence. Consequently, the suggested new modified flow shop scheduling algorithm is prepared to decrease, and this algorithm is capable of minimizing TCT. So far, only the makespan objective is analyzed in the literature and we are the first to analyze TCT.

REFERENCES

1. *Baker K R (1974), Introduction to Sequence and Scheduling, wiley.*
2. *Gupta J N D (1971), "A functional heuristic for the flow shop scheduling problem", Operational Research Quartely, **22**, pp 39-47.*
3. *Johnson S M (1954), "Optimal two and three stage production schedules with setup times included", Naval Research Logist Quartely, **1**(1) ,pp 61-68.*
4. *Jayakumar S, Meganathan R and Sathiya Shanthi R (2016), "An Heuristic Approach for Solving two machine n-job flow shop scheduling problem with makespan objective" ,IOSR Journal of Mathematics, **12**(3) , pp 23-26.*
5. *Jayakumar Sundaramoorthy, Meganathan Rangaraji, Sathiya Santhi (2019), "An Heuristic Approach for Solving permutation flow shop scheduling problem", International Journal of Emerging Technology and advanced Engineering, **6**(4) , pp 144-146.*
6. *Jayakumar S and Vasudevan G (2019), "Heuristic algorithm for permutation flow shop scheduling", International Journal of Technology, **9**(1), pp 5-7.*
7. *Jayasankari S, Jayakumar S and Vijayaragavan R (2021), "An efficient Flow shop Scheduling Problem with Makespan Objective" Turkish Journal of Computer and Mathematics Education, **12**(4) ,pp 461-466.*
8. *Jayakumar S and Meganathan R (2014),"Heuristic approach for OSSP to minimize makespan with release dates", Industrial Engineering Journal vol.7, pp 17-22*
9. *Vasudevan G, Jayakumar S and Sathya Shanthi R (2020), "A better heuristic approach for n-job m- machine flow shop scheduling problem", Journal of Computational Mathematica, **4**(1) , pp 118-124.*
10. *Vasudevan G and Jayakumar S (2020), "The heuristic algorithm for flow shop scheduling problem with TET objective", Malaya Journal of Matematik, **8**(4) , pp2071-2073.*
11. *Vasudevan G and Jayakumar S (2020), "An algorithm for solving n-job m-machine flow shop scheduling problem with objective of minimizing makespan", International Journal of Advanced Research in Engineering and Technology (IJARET), **11**(11) , pp1203-1207.*

APPENDIX

1.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	3	2	4
J2	1	3	1
J3	2	2	3
J4	1	4	2

2.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	2	5	7
J2	4	6	8
J3	3	1	9
J4	7	4	3

3.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	3	2	1
J2	6	2	5
J3	4	5	3
J4	7	6	4

4.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	3	1	2
J2	5	3	4
J3	1	5	6
J4	3	7	2
J5	6	3	4

5.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	2	6	4
J2	3	4	5
J3	2	3	4
J4	6	4	2
J5	2	2	7

6.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4
J1	2	5	6	5
J2	4	6	7	2
J3	3	3	2	3
J4	2	2	3	6
J5	1	4	4	1

7.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4
J1	5	4	5	3
J2	6	2	1	6
J3	2	7	1	3
J4	3	6	2	4
J5	4	3	4	5

8.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4
J1	2	3	3	7
J2	2	4	2	3
J3	3	6	2	4
J4	2	6	4	6
J5	5	2	3	6
J6	1	2	1	3

9.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4
J1	6	5	4	3
J2	5	4	3	6
J3	4	3	5	5
J4	3	2	6	6
J5	2	1	5	6
J6	1	6	4	4

10.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4	M/C/5
J1	4	9	8	7	5
J2	6	4	6	5	6
J3	5	6	7	6	4
J4	7	7	5	4	3
J5	8	5	4	3	9
J6	9	8	3	9	7

11.

Job/Machine	M/C/1	M/C/2	M/C/3	M/C/4	M/C/5
J1	2	5	4	2	5
J2	6	3	7	3	6
J3	5	4	6	5	4
J4	3	6	3	2	3
J5	4	2	5	4	4
J6	2	5	8	7	5

12.

Job/Machine	M/C/1	M/C/2	M/C/3
J1	2	1	4
J2	3	4	5
J3	5	5	6
J4	5	6	4
J5	6	3	5
J6	7	2	6
J7	3	5	7

AUTHORS

R Durgadevi, Research Scholar, PG and Research, Department of Mathematics, AAGA College, Cheyyar - 604 407, Tamil Nadu, India

E-Mail: durgadevi.apagri@gmail.com

Dr. S. Jayakumar, Assistant Professor & Head, PG and Research, Department of Mathematics, AAGA College, Cheyyar – 604 407, Tamil Nadu, India.
E-Mail: sundarajayakumar@gmail.com

K Moorthi, Research Scholar, PG and Research, Department of Mathematics, AAGA College, Cheyyar - 604 407, Tamil Nadu, India.
E-Mail: m.s.kiruthilaya@gmail.com