An Jayvasu Algorithm For Solving Permutation Flow Shop Scheduling Problem

G. Vasudevan and S. Jayakumar Sundramurthy

October 21, 2020

An Jayvasu Algorithm For Solving Permutation Flow Shop Scheduling Problem

G. VASUDEVAN[1] Dr. S. JAYAKUMAR[2]
1. go.vasudevan@gmail.com, 9600436025.
Research Scholar PG & Research Department of Mathematics, Arignar Anna Government Arts College, Cheyyar, Tamil Nadu, India.
2. sundarajayakumar@gmail.com, 9445250963.
Assistant Professor & Head, PG & Research Department of Mathematics, Arignar Anna Government Arts College, Cheyyar, Tamil Nadu, India.

Abstract: In this paper, we have focused our attention on minimizing the total elapsed time for solving flow shop scheduling. We have given a simple heuristic approach namely **Jayvasu** algorithm to solve the permutation flow shop scheduling problem. The analysis and result indicate that our algorithm performs better than the algorithms available in the literature namely Palmer's, CDS, Gupta, RA, and NEH. In fact, our algorithm is simple and easy to use when compared with others.

Keywords: Flow shop, Heuristic, total elapsed time, Scheduling.

1. INTRODUCTION

The sequencing problem deals the determination of an optimal sequence of jobs by a number of facilitiesso as to optimize the total elapsed time. Sequencing problems may be classified into two categories:

In the first category, there are *n* different jobs to be carryout, where all the job need processing on some or all of m different types of machines. The order in which these machines are to be used for processing each job (for example, each job is to be processed first on machine A, then on B and thereafter on C i.e., in the order ABC) is given. Also, the expected or actual processing time of each job on each machine is known. We can also decide the effectiveness for any given sequence of jobs at each of the machines and we wish to select from the $(n!)^m$ *theoretically feasible* alternatives, the one which is both technologically feasible and optimizes the effectiveness measure (e.g. minimizes the total elapsed time from the start of the first job on the first machine to the completion of the last job on the last machine as well as idle time of machines). A technologically feasible sequence is one which satisfies the constraints (if any) on the order in which each job must be performed through the *m* machines. The technologically of manufacturing processesrenders many sequence technologically infeasible. For example, a part must be degreased before it is painted; similarly, a hole must be drilled before it is threaded.

Although, theoretically, it is always possible to select the best sequence by testing each one; in practice, it is impossible because of the large number of computations involved. For example, if there are 4 jobs to be processed on each of the 5 machines (*i.e.* $n = 4$ and $m = 5$), the total number of theoretically possible different permutation sequences will be $(4!)^5 = 7,962,624$. Of course, as already said, some of them may not be feasible because the required operations must be performed in a specified order. Obviously, any technique which helps us arrive at an optimal (or at least approximately so) sequence without trying all or most of the possibilities will be quite valuable.

The second category of problems deals with job shops having a number of machines and a list of tasks to be performed. Each time a task is completed by a machine, the next task to be started on it has got to be decided. Thus the list of tasks will change as fresh orders are received.

Unfortunately, both type of problems are intrinsically difficult. While solutions are possible for some simple cases of the first, only some empirical rules have been developed for the second type till now.

1.1 SEQUENCING PROBLEMS

In sequencing problems, there are two or more customers to be served (or jobs to be done) and one or more facilities (machines) are available for this purpose. We want to know when each job is to begin and what its due date is. We also want to know which facilities are required to do each job, in which order these facilities are required and how long each operation is to take.

Sequencing problems have been most commonly encountered in '**production shops**' where different products are to be processed over various combinations of machines.

It may be noted the sequencing problems can arise even one service facility is involved. Since the total elapsed time (also called makespan) is fixed and equal to the sum of processing times for all jobs for all possible sequences, some other optimality criteria *e.g.*, minimizing the mean flow time or some function of the lateness of the jobs may be applied in such problems.Lateness of a job is defined asthe difference between the actual completion time of the job and its due date. If lateness is positive, it is termed as **tardiness**. Total tardiness is the sum of tardiness over all the jobs in the set.

The various optimality criteria normally resorted to are:

1. Minimizing total elapsed time (or makespan).
2. Minimizing mean flow time (or mean time in the job shop).
3. Minimizing idle time of machines.
4. Minimizing total tardiness.
5. Minimizing number of tardy jobs.

6. Minimizing in-process inventory cost.
7. Minimizing the cost of being late.

A general sequencing problem may be defined as follows:

Let there be *n* jobs (1, 2, 3…*n*), each of which has to be processed, one at a time, on each of the *m* machines (A, B, C …). The order of processing each job through the machines is given (for example, job 1 is processed on machines A, C, B, in this order). Also, the time required for processing each job on each machine is given. The problem is to find among $(n!)^M$ possible sequences, that *technologically feasible* sequence for processing the jobs which gives the *minimum total elapsed time* for all the jobs.

Symbolically,

Let A_i = time required for job *i* on machine A,

B_i = time required for job *i* on machine B, etc., and

T = total elapsed time for jobs 1, 2, 3….n *i.e.*, time from start of the first job to completion of the last job.

The problem is to determine a sequence $(i_1, i_2, ..., i_n)$ where $(i_1, i_2, ..., i_n)$is a permutation of integers (1, 2, …, *n*) which will minimize T.

Analytic methods have been developed for solving only five simple cases:

(1) *n* jobs and one machine A.

(2)*n* jobs and two machines A and B; all jobs processed in the order, say AB.

(3)*n* jobs and three machines A, B and C; all jobs processed in the order , say ABC.

(4) two jobs and *m* machines ; each job to be processed through the machines in a prescribed order , not necessarily the same for both jobs.

(5) *n* jobs and *m* machines A, B, C, …, K; all jobs processed in the order, say ABC … K.

2. LITERATURE REVIEW

One of the prior Heuristic Johnson's algorithm(1954) consider for the two machine flow shop problem with objective of minimizing total elapsed time after that in 1965 Palmer proposed a Heuristic algorithm which is a slope order index to sequence of jobs on the machines based on the processing time and known as the Palmer's Heuristic for the flow shop scheduling problem. It was to give priority job so that jobs with processing times that tends to increase from machine to machines will receive higher priority for flow shop scheduling problem.

Campbell, Dudek and Smith (CDS) (1970) proposed a Heuristic that was extension of Johnson's Algorithm for a FSSP with total elapsed time minimization.

Gupta [1971] suggested another heuristic which was similar to Palmer's heuristic. He defined the slope index in a different manner by taking into account some attractive facts about optimality of Johnson's rule for the three machine problems.

Dannenbring [1977] developed a Heuristic algorithm called RA (rapid access) which combines the merits of Palmers Heuristic (slope index) and the CDS algorithms. Its result is to give a best solution as fast and simply as feasible. As an alternative of solving m-1 artificial two machine problems, it solves only one artificial problem using Johnson's algorithm(1954) in which the processing times are decide from a stand by scheme.

The Nawaz, Enscore, and Ham (NEH) [1983] heuristic algorithm is based on the assumption that a job with high total processing time on all the machines should be given higher priority than job with low total processing time. The NEH algorithm does not transform that original m-machine problem in to one artificial two-machine problem. It builds the final sequence in a constructive way, adding a new job at each step and finding the best partial solution. Jayakumar et al(2016) solved the two machine n job flow shop scheduling problem with makespan objective. Later on Jayakumar et al(2019) solved the permutation flow shop scheduling problem with a heuristic approach. In this paper also a simple Heuristic approach is used to solve the flow shop scheduling problem with the objective of minimizing the total elapsed time.

3. ASSUMPTIONS IN SEQUENCING PROBLEMS

The following simplifying assumptions are usually made while dealing with sequencing problems:

(*i*) Only one operation is carried out on a machine at a particular time.

(*ii*) Each operation, once started, must be completed. i.e., pre-emption not allowed

(*iii*) An each operation must be completed before its succeeding operation can start.

(*iv*)Only one machine of each type is available.

(*v*) A job is processed as soon aspossiblebut onlyin the order specified.

(*vi*)Processing the time are independent of order of performing the operations i.e., No parallel processing

(*vii*) The transportations time *i.e.*,the time required to transport jobs from one machine to another is negligible.

(*viii*)Jobs are completely known and are ready for processing when the period under consideration starts.

4. Jayvasu Algorithm

For the n job m machine case, with the objective of minimizing total elapsed time, Johnson's algorithm fails when both condition that of Minimum of M1 is greater than or equal to maximum of M2, M3, … $M_{m-1}$and Minimum of M_m is greater than or equal to maximum of M2, M3, … M_{m-1}under such circumstances Heuristic plays a predominant role. Here we have developed one model using Heuristic approach which yield excellent output with in a quick time rather than using NEH, CDS Algorithms. Our algorithm has been compared with Palmer, CDS, Gupta, RA, NEH algorithms and yieldsbetter result using the concept of Johnson's algorithm after converting it to **Jayvasu** algorithm.

4.1 Steps of our Algorithm

Step 1: Assume rows are jobs and columns are machines

Step 2: Find the maximum processing times on each columns.

Step 3:More than two columns have maximum and same number of processing times. This is tie for select most two.

Step 4: Make all possible pair of columns(Machines) to consider as two machine problem.

Step 5: Using John's Algorithm to find optimum sequence of all the sequences.

4.2 Numerical problem

.	**M1**	**M2**	**M3**	**M4**	**M5**
J1	3	2	1	4	5
J2	6	2	5	3	4
J3	4	5	3	1	2
J4	7	6	4	2	3
J5	3	5	1	3	6

Problem solved using **Jayvasu** Algorithm

Maximum processing times of columns are 7, 6, 5, 4, 6.

Choose two machines associated with the most processing time.

Here 7 in M1 and 6 in M2 and M5, the possible pairs are M1& M2, M1&M5. Now using Johnsons algorithm we get the total elapsed time associated with the sequence

The total elapsed time of M1&M2 is 41(J5,J3,J4,J2,J1) and 39(J5,J3,J4,J1,J2)

The total elapsed time of M1&M5 is 36(J1,J5,J2,J4,J3) and 36(J5,J1,J2,J4,J3)

5. Result Analysis

Comparison of the result using Jayvasu and others such as Palmer, CDS, NEH, RA and Gupta

No. Of observations	Technique	Optimal sequence	Total elapsed time	**No. of alternative sequence**
01	Palmer	J1,J5,J2,J3,J4	38	1
02	CDS	J1,J2,J5,J4,J3 J1,J5,J2,J4,J3	36	2
03	Gupta	J1, J3, J5, J2, J4	38	1
04	RA	J1,J5,J2,J4,J3	36	1
05	NEH	J1,J5,J2,J4,J3	36	1
06	**Jayvasu**	J1,J5,J2,J4,J3 J5,J1,J2,J4,J3	**36**	**2**

Using **Jayvasu** algorithm we have got two alternative near optimal sequence having the same total elapsed time whereas in other algorithms such as CDS gives only 2 alternative sequence with the same total elapsed time and RA & NEH gives only one alternative sequence and Gupta and Palmer yield total elapsed time as 38 which is higher than our algorithm.

6. CONCLUSION

Based on the result obtained, it has been found that **Jayvasu** algorithm yields better result in simple method compare to other algorithm found in the literature. We conclude that whenever Johnson method fails to solve n job m machine case our algorithm is suitable one to solve when compared to others.

7. REFERENCES

1. Baker, K. R (1974). Introduction to Sequence and Scheduling, Wiley.
2. Campbell, H. G., Dudek, R. A. and Smith, M. L. (1970). A heuristic algorithm for the n-job, m-machine sequencing problem. Management Science, 16, B630-B637.
3. Dannenbring D. (1977). An evolution of flow shop scheduling heuristics. Manage Science, Vol. 23, Issue 11, pp.1174-1182.
4. Gupta, J.N.D. (1971). A functional heuristic for the flow-shop scheduling problem, Operational Research Quarterly, 22, 39-47.
5. Jayakumar S, SathiyaShanthi Ramesh Babu, MeganathanRangaraji,(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, Meganathan R, SathiyaShanthi , (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.
7. Johnson S. M. (1954). Optimal two and three stage production schedules with setup times included, Naval Research Logist Quarterly, Vol. 1, Issue 1, pp.61-68.
8. Nawaz, M., Enscore, E. E., and Ham, I., (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. OMEGA, 11, pp 91-95.
9. Palmer D. S. (March 1965). Sequencing jobs through a multi stage process in the minimum total time – a quick method to obtaining a near optimum. Operational research quarterly, Volume 16, no.1, pp101-107.
10. Sahu A K. (2009). Efficient heuristics for scheduling tasks on a flow shop environment to optimize make span National Institute of Technology, Rourkela INDIA.