



FLOWSHOP SCHEDULING USING A NETWORK APPROACH

M. H. Oladeinde^{1,*}, A. I. Momodu² and C. A. Oladeinde³

^{1,2,3}DEPARTMENT OF PRODUCTION ENGINEERING, FACULTY OF ENGINEERING, UNIVERSITY OF BENIN, NIGERIA.

*E-mail addresses:*¹ moladeinde@uniben.edu,² oloagbaye.momodu@uniben.edu,³ kallayama@yahoo.com

ABSTRACT

In this paper, a network based formulation of a permutation flow shop problem is presented. Two nuances of flow shop problems with different levels of complexity are solved using different approaches to the linear programming formulation. Key flow shop parameters inclosing makespan of the flow shop problems were obtained without recourse to the traditional approach of using Gantt charts. The linear programming models of the flow shop problems considered were solved using LINGO 7.0. The present technique has been shown to be very effective and efficient.

Keywords: Flow shop, network, linear programming, makespan, Gantt Chart, LINGO

1. INTRODUCTION

The traditional flow shop scheduling problem, in which a finite number of jobs require processing on a specified number of machines, is a fundamental production scheduling research problem. Each unique job consists of a number of tasks and is processed by routing it through each machine in a prescribed order. The objective is to find a permutation of jobs which minimizes some objective of interest. The most frequently studied objective is makespan, which is the time when the last job completes on the last machine. The objective may be machine related or job related. Minimization of the overall completion times of all the jobs in the flow shop is the most common objective documented in literature. A complete survey of the flow shop scheduling problems has been reported by [1]. Minimizations of both tardiness and flow time of jobs have also been reported in literature.

The flow shop scheduling problem has been shown to be NP-hard by [2]. A number of nuances of the flow shop scheduling problem exist. In some instances, all the jobs in the flow shop are processed through the machine in the same order. This constitutes a subclass of problems known as permutation flow shop problem. From an application stand point, one or more jobs in a permutation flow shop scheduling problem may not require processing in one or more machines resulting in what is known as a bypass flow shop scheduling problem. A hybrid flow shop is different from the traditional flow shop in the sense

that it consists of several manufacturing stages with several unrelated parallel machines instead of a sequence of single machines [3]. An extensive review on research carried out on Hybrid Flow Shop has been presented by [4].

A number of techniques [5] have been proposed to obtain near optimal solutions to flow shop scheduling problems. They include genetic algorithm ([6], [7]), Ant Colony Optimization [8], NEH algorithm [9], CDS algorithm [10], Simulated Annealing [11], and Tabu Search [12]. The different exact techniques and heuristics for flow shop scheduling can only be used to find the sequence of the jobs which optimizes the objective of interest but does not provide the value of the objective. The Gantt chart is commonly used for obtaining the objective value after the optimal sequence has been obtained. A Gantt chart is a bar chart that is a visual representation of the sequencing and duration of activities on any given project [13]. The use of Gantt charts for presenting the start and finish time of jobs on different machines as well as the completion time of all jobs can become complicated to develop and understand when the number of jobs and machines increases. To address this challenge, a network representation of a flow shop scheduling problem is developed to facilitate the determination of the makespan of a permutation flow shop. The developed network representation of the flow shop is solved using integer linear programming approach.

2. FLOW SHOP SCHEDULING

A flow shop scheduling problem involves scheduling n jobs on m machines. A job consists of m tasks or operations and the j^{th} operation of each job must be processed on the j^{th} machine. A job can only be processed on machine j if it has been completely processed on machine $j-1$ and machine j is idle. A job j_i can be viewed as a set of tasks, with each of the tasks carried out on a specific machine. The time it takes to process a job on a machine may be constant, non-negative or even zero. In the case where the processing time of a job on a machine is zero, it implies that the job is not processed on the machine in question. A number of assumptions are made in flow shop scheduling. All jobs are ready to be processed at time zero, each job has a predetermined operation time on all machines, machines can be kept idle, always available and can process only one operation at a time, jobs are non preemptable and set up times of jobs on the machines are sequence independent and the set up times as well as transportation times of a job are included in the processing time for jobs. Consider the following sequence of jobs in a flow shop: $J_i = \{J_{i1}, J_{i2}, \dots, J_{im}\}$ and p_{ij} = processing time of task i on machine j . According to the nomenclature, the permutation flow shop is depicted as $F/Permu/C_{max}$, where the objective to be minimized is the makespan.

3. CONVERTING A FLOW SHOP TO A CPM NETWORK

In general, the procedure for generating the precedence relationships amongst the different tasks of the jobs can be summarized below. Consider a flow shop with n jobs and m machines. A job i consists of m tasks, namely, $J_{i1}, J_{i2}, \dots, J_{im}$. Let the position the job occupies be represented by a vector $P = (1, 2, 3, \dots, n)$. Let m represent the number of machines in the flow shop, which is equal to the number of tasks required to completely process a job:

1. The first task for the first Job (Job 1) on machine 1 has no predecessor.
2. The first task for the first Job (Job 1) on machine $m > 1$ is preceded by the first task of the first Job (Job 1) on machine $m-1$ or the $(m-1)$ th machine.
3. The first task for every Job $n > 1$ on machine $m=1$ is preceded by the first task on Job $n-1$.

4. Every other task of a Job $n > 1$ for machine $m > 1$ has two predecessors the task of Job $(n-1)$ on machine m and the task of Job n on machine $m-1$.

4. LINEAR PROGRAMMING

Linear programming is an optimization technique. It is used in Operations research to find optimal values of decision variables for problems which can be represented in form of objective function(s) and constraints. The linear programming technique can be used to find the longest path in the CPM network formulated from the flow shop problem. The longest path corresponds to the critical path in the network. Two approaches are available for critical path analysis of project networks. The choice of the formulation approach depends on the component(s) of the output of the solution that is/are of interest to us. However, in the present paper, we present the two methods of formulation and illustrate each with a numerical example.

5. LONGEST PATH FORMULATION

Assume that a unit flow enters a project network at the source (start node) and leaves at the sink (finish node) of the project network. Given that there are n nodes in the network, we adopt the following notation. x_{pq} = A binary variable which represents the amount of flow through a path between nodes p and q in a project network. If the value of the variable assumes 1, it implies that there is a unit flow through the path and 0 otherwise. The duration of task between nodes p and q is given as t_{pq}

The longest path of the network can be found using the following integer LP formulation

$$Max = \sum_{p=1}^n \sum_{q=1}^n t_{pq} x_{pq} \tag{1}$$

$$\text{Subject to } \sum_{q=1}^n x_{1q} = 1 \tag{2}$$

$$\sum_{q=1}^n x_{pq} = \sum_{k=1}^n x_{kp} \quad p = 2, \dots, n-1 \tag{3}$$

$$\sum_{k=1}^n x_{kn} = 1 \tag{4}$$

$$x_{pq} = 0 \text{ or } 1$$

In the linear programming model described in Eqs 1 – 4, Eq (1) ensures that the unit flow goes through the longest path in the network; Eq (2) guarantees that

only a unit flow enters the network through the source (starting node); Eq (3) forces conservation of flow at internal node in the project network. Specifically, it ensures that a unit of flow goes into a node and a unit of flow leaves the node.

6. LINEAR PROGRAM MODEL

Consider a project network with activity time t_{ij} where the vertices (V) represent the set of node numbers and the direct edges represent the activities. An activity is represented by one and only one arrow with its tail event at node i and its head event at node j ($i, j \in V$) where $i < j$. Let x_j denote the earliest time that the event corresponding to j occurs. Since, $i < j$, it implies that for each pair of directed edge (i, j) , $x_j \geq x_i + t_{ij}$. Assuming the project has n nodes, our objective is to minimize the time required to complete the project. The linear program to find the earliest event times and the project completion time can be stated formally as:

Minimize $x_n - x_1$

(5)

Subject to $x_j \geq x_i + t_{ij} \quad \forall (i, j) \in A$

(6)

$x_i, x_j \geq 0 \quad \forall i \in V$

In the linear program, A and V represent the set of directed edges and nodes respectively. In the linear

programming model presented, the critical activities correspond to the constraints which are satisfied as equations at the optimal solution.

7. NUMERICAL SOLUTION

In this section, the applications of the two approaches for scheduling flow shop problems are presented. In the first example, our interest is to find the makespan of the flow shop whose optimal sequence has been obtained a priori. The second example illustrates the case where we are also interested in the start, finish, job waiting and machine idle times in addition to the makespan of the flow shop problem.

8 NUMERICAL EXAMPLE 1

In the following 7 job 2 machine problems, the processing times (hours) for each job on each machine is presented in Table 1:

Table 1: Jobs and processing times of example 1

Job	A	B	C	D	E	F	G
Machine 1	6	24	30	12	20	22	18
Machine 2	16	20	20	13	24	2	6

The optimal sequence which minimizes the makespan of the flow shop problem is A-D-E-C-B-G-F. The optimal sequence is obtained using Johnson's rule. Johnson's rule gives the optimal sequence but not the makespan of the flow shop problem. Using the methodology described previously, the following network is developed from the flow shop problem.

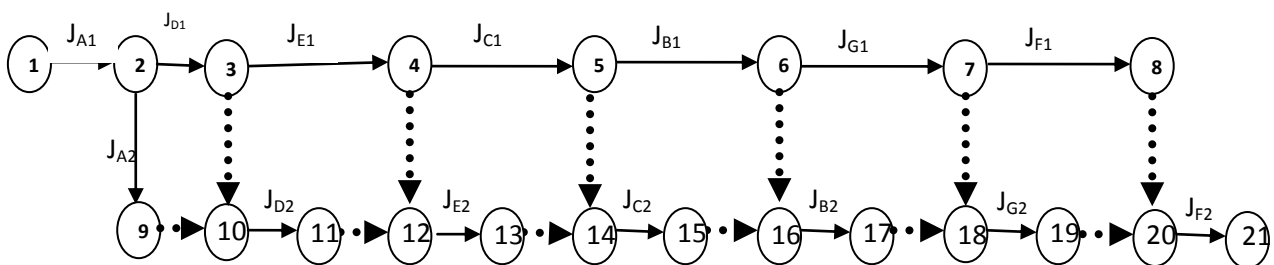


Figure 1: Network diagram of flow shop problem

The linear program in LINGO is shown below:

(Objective function)

max=6*x12+12*x23+20*x34+30*x45+24*x56+18*x67+22*x78+16*x29+13*x1011+24*x1213+20*x1415+20*x1617+6*x1819+2*x2021+0*(x310+x910+x1112+x412+x1314+x514+x1516+x616+x1718+x718+x1920+x820);

(network constraints)

x12=1;x12=x23+x29; x23=x34+x310; x29=x910; x910+x310=x1011; x34=x45+x412; x1011=x1112; x1112+x412=x1213; x45=x56+x514; x1213=x1314; x1314+x514=x1415; x56=x67+x616; x1415=x1516; x1516+x616=x1617; x67=x78+x718; x1617=x1718;

$x_{1718} + x_{718} = x_{1819}$; $x_{78} = x_{820}$; $x_{1819} = x_{1920}$; $x_{1920} + x_{820} = x_{2021}$; $x_{2021} = 1$;
 (binary constraints)

$@bin(x_{12})$; $@bin(x_{23})$; $@bin(x_{34})$; $@bin(x_{45})$; $@bin(x_{56})$; $@bin(x_{67})$; $@bin(x_{78})$;
 $@bin(x_{29})$; $@bin(x_{910})$; $@bin(x_{1011})$; $@bin(x_{1112})$; $@bin(x_{1213})$; $@bin(x_{1314})$; $@bin(x_{1415})$; $@bin(x_{1516})$;
 $@bin(x_{1617})$;
 $@bin(x_{1718})$; $@bin(x_{1819})$; $@bin(x_{1920})$; $@bin(x_{2021})$;

The solution to the network problem is shown in Table 2

Table 2: Solution to example 1

Variable	Value
X12	1
X23	1
X34	1
X45	1
X56	1
X67	1
X78	1
X29	0
X1011	0
X1213	0
X1415	0
X1617	0
X1819	0
X2021	1
X310	0
X910	0
X1112	0
X412	0
X1314	0
X514	0
X1516	0
X616	0
X1718	0
X718	0
X1920	0
X820	1

The objective function value is equal to 134

9. NUMERICAL EXAMPLE 2

Consider the following five jobs three machines flow shop scheduling problem. The optimal sequence which minimizes the makespan has been found using Johnson’s rule and is ADEBC. The task is to determine the makespan of the flow shop

Table 3: Jobs and processing times of example 2

Job	A	B	C	D	E
Machine 1	4	9	8	6	5
Machine 2	5	6	2	3	4
Machine 3	8	10	6	7	11

The flow shop problem with the sequence ADEBC is equivalent to the network model shown in Figure 2.0.

The project network data for the numerical example is presented in Table 4.

Table 4: Project data for network model of flow shop

Node	Activity(job)	Predecessor	Duration
1-2	JA1	-	4
2-3	JA2	JA1	5
2-4	JD1	JA1	6
3-8	JA3	JA2	8
5-7	JD2	JA2, JD1	3
4-6	JE1	JD1	5
9-10	JE2	JE1, JD2	4
8-11	JD3	JD2, JA3	7
6-12	JB1	JE1	9
13-14	JB2	JB1, JE2	6
11-15	JE3	JE2, JD3	11
13-16	JC1	JB1	8
17-18	JC2	JC1, JB2	2
15-19	JB3	JB2, JE3	10
20-19	JC3	JC2, JB3	6

The network model of the flow shop problem is shown in Figure 2.

The jobs in the flow shop in the respective machines have been represented as activities on the project network. An activity is denoted by a directed line between two nodes. The activities have been carefully ordered in the network diagram to retain the technological constraints. The uses of dummy activities have been used to preserve the precedence requirements amongst the different activities in the project network. Dummy activities have been denoted by directed dashed lines. Based on the developed network diagram, a linear programming model was formulated for the flow shop problem using the method adopted by [14]. The decision variables used in the formulation of the linear program are the start and finish times of the activities (jobs on machines). The linear program is presented here under in Figure 2.

The linear program was run using LINGO 7.0. A global optimum result was obtained along with the associated values of the decision variables. The output of LINGO 7.0 for the twenty (20) decision variables is presented in Table 5.

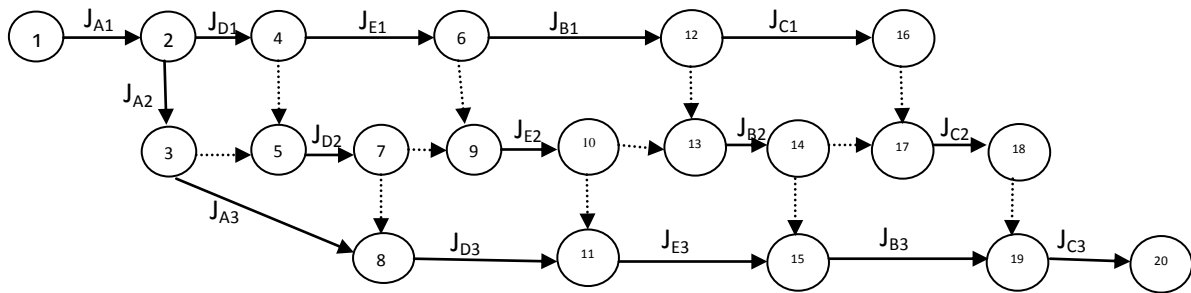


Figure 2: Network representation of flow shop problem

(objective function)

$$\text{MIN} = T_{20};$$

(network constraints)

$$T_2 - T_1 \geq 4; T_4 - T_2 \geq 6; T_6 - T_4 \geq 5; T_{12} - T_6 \geq 9; T_{16} - T_{12} \geq 8; T_3 - T_2 \geq 5;$$

$$T_5 - T_4 \geq 0; T_9 - T_6 \geq 0; T_{13} - T_{12} \geq 0; T_{17} - T_{16} \geq 0; T_5 - T_3 \geq 0; T_7 - T_5 \geq 3;$$

$$T_9 - T_7 \geq 0; T_{10} - T_9 \geq 4; T_{13} - T_{10} \geq 0; T_{14} - T_{13} \geq 6; T_{17} - T_{14} \geq 0; T_{18} - T_{17} \geq 2;$$

$$T_8 - T_3 \geq 8; T_{11} - T_8 \geq 7; T_{15} - T_{11} \geq 11; T_{19} - T_{15} \geq 10; T_{20} - T_{19} \geq 6; T_8 - T_7 \geq 0;$$

$$T_{11} - T_{10} \geq 0; T_{15} - T_{14} \geq 0; T_{19} - T_{18} \geq 0;$$

Table 5: solution to example 2

Variable	Value
T20	51
T2	4
T1	0
T4	10
T6	15
T12	24
T16	32
T3	9
T5	10
T9	15
T13	24
T17	32
T7	13
T10	19
T14	30
T18	34
T8	17
T11	24
T15	35
T19	45

The result of the flow shop scheduling problem is summarized in Table 6. The Table shows the start, finish and the waiting times of the jobs on each machine in the flow shop problem.

Table 6: Summary of result of numerical example 2

Machine	Job	Start	Finish	Job waiting time	Machine Idle time
Machine 1	A	0	4	0	0
	D	4	10	4	
	E	10	15	10	
	B	15	24	15	
Machine 2	C	24	32	24	10
	A	4	9	0	
	D	10	13	0	
	E	15	19	0	
Machine 3	B	24	30	0	0
	C	32	34	0	
	A	9	17	0	
	D	17	24	4	
Machine 3	E	24	35	5	0
	B	35	45	5	
Machine 3	C	45	51	11	0
	C	45	51	11	

10. DISCUSSION OF RESULTS

The result shows for numerical example I shows that a unit flow passes through the paths 1-2-3-4-5-6-7-8-20-21. The variables representing paths which connect the pair wise node numbers assumed a value of unity from the solution to the linear programming model. The addition of the duration of the tasks along this path amounts to 134 hours. Therefore the makespan of the flow shop problem for the sequence ADECBGF is equal to 134 hours.

Table 6 shows a summary of the result extracted from the LINGO 7.0 output for numerical example 2. The

computation reveals that the makespan is equal to 51 minutes. This is equal to the event time of the terminal node in the project network. As expected for a permutation flow shop problem, there is no job waiting time on the first machine in the flow shop. The start and finish times of the jobs on the machines shows that a total of 10 minute idle time is obtained on machine 2 while no inserted idle times are included for the third machine.

11. CONCLUSION

A network methodology has been applied flow shop problem. Linear programming has been applied to the network equivalent of two nuances of flow shop problem. The method has facilitated the determination of key flow shop parameters without construction of Gantt chart. The current method is inherently flexible as it allows changes in problem parameters to be handled efficiently.

12. REFERENCES

[1] Hejazi S. R. and Saghafian S, "Flow shop Scheduling problems with makespan criterion: A review", *International Journal of Production Research*, Vol. 43, No. 14, 2005, pp. 2895-2929.

[2] Kise, H, "On an automated two-machine Flowshop scheduling problem with infinite buffer", *J. Oper. Res. Soc. Japan* 34, 1991, pp 354-361.

[3] Havill, J.T and Mao,W, "On-line Algorithms for Hybrid Flow Shop Scheduling", *Proceedings of the Fourth Joint Conference on Information Sciences, Research Triangle Park, North Carolina*, 1998, pp. 134-137.

[4] Riuz, .R. and Vazquez-Rodriguez, J.A. "The Hybrid Flow Shop Scheduling Problem", *European Journal of Operational Research*, 205(1),2010, pp 1-18.

[5] Onyeagoro, E.O, " Sequencing for Batch Production in a Shop Flowline Machine Shop", *Nigerian Journal of Technology*, Vo 6, Number 1, 1982, pp. 16 - 24.

[6] Ruiz, R and Maroto. C. "A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility", *European Journal of Operational Research* 169 (3); 2006,pp 781-800.

[7] Reeves, C.R "Genetic Algorithms, Path Relinking and the Flowshop Sequencing Problem *Evolutionary Computation Journal*, Vol.6 No.1, 1998, pp. 230-234

[8] Shyu, S.J and Lin, B.M.T and Yin, P.Y "Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time", *Computers and Industrial Engineering*, Volume 47, issue 2-3, 2004, pp 181 - 195.

[9] Nawaz, M., Enscore Jr, E. and Ham, I., "A heuristic algorithm for them-machine,n-job flow-shop sequencing problem".*Int. J. Manag. Sci.*, 11, 1983, pp. 91-95.

[10] Campbell, H.G., Dudek, R.A. and Smith, M.L,"A heuristic algorithm of then-job,m-machine sequencing problem", *Manag. Sci.*, 16, 1970, pp. 630-637.

[11] Osman, I.H and Potts, C.N "Simulated annealing for permutation flow-shop scheduling", *Omega* vol. 17, issue 6, 1989, pp 551-557.

[12] Hooda N, and Dhingra A. K "Flow Shop Scheduling using Simulated Annealing: A Review", *International Journal Of Applied Engineering Research*,Volume 2, No 1,2011, pp 234 - 249.

[13] Elmabrouk, O.M "A Linear Programming Technique for the Optimization of the Activities in Maintenance Projects", *International Journal of Engineering and Computing*, vol 11, no 1, 2011, pp 24 - 29.

[14] Oladeinde M.H and Oladeinde.C.A, "Modeling Critical Path Analysis using linear programming", *Journal of Mathematics and Technology*, 3(2), 2012, pp. 10 - 14.