Journal of Project Management 8 (2023) 45–56

Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

An ordered precedence constrained flow shop scheduling problem with machine specific preventive maintenance

T. Jayanth Kumar[a] and M. Thangaraj[a*]

[a]*Department of Mathematics, Faculty of Engineering and Technology, Jain University, Bangalore, Karnataka, India*

C H R O N I C L E

Article history:
Received: June 11, 2022
Received in revised format: July 22, 2022
Accepted: August 12, 2022
Available online:
August 12, 2022

Keywords:
Flow shop-scheduling problem
Heuristic algorithm
Preventive maintenance

A B S T R A C T

In reality, the machines may interrupt because of the nature of deterioration of the machines. Thus, it is inevitable to perform maintenance alongside production planning. The preventive maintenance is a schedule of strategic operations that are performed prior to the failure occurring, to retain the system operating at the preferred level of consistency. Thus, preventive maintenance plays a significant role in flow shop scheduling models. With its practical significance, this study addresses a practical three-machine *n* jobs flow shop-scheduling problem (FSSP) in which machine specific preventive maintenance, where each machine is given with a maintenance schedule is considered. In addition, a practical ordered precedence constraint in which some set of jobs has to process in the specified order irrespective of their processing times is also considered. The problem's goal is to establish the optimal job sequence and preventive maintenance such that the overall cost of tardiness and preventive maintenance is as minimum as possible. An efficient heuristic approach is designed to tackle the present model, resulting in total cost savings. A comparative analysis is not conducted due to absence of studies on the current problem in the literature. However, Computational experiments are carried out on some test instances and results are reported. The reported results may be useful for future studies.

1. Introduction

The flow shop-scheduling problem (FSSP) is one of the most hard decision problems and has been considered by several researchers over the past few decades. In a classical FSSP, machines are placed in a sequence, and jobs has to go through the machines in the same order, i.e. each job has operations, and the k^{th} operation of all jobs is handled on machine k for $k = 1,\ldots,m$ (Lee & Kim, 2017). Here, an operation is the process of performing a job on a machine. Because the flow shop diversified practical applications in the manufacturing process; the literature witnessed several variants and constrained versions of classical FSSP in the past few decades. To cite few, FSSP with set up times (Allahverdi & Al-Anzi, 2006), FSSP with preventive maintenance (Ruiz et al., 2007), distributed permutation FSSP (Gao & Chen, 2011), FSSP with known breakdown times and job weights (Baskar & Xavior, 2012), FSSP with machine and job priorities (Baskar & Xavior, 2014), distributed permutation FSSP with setup times (Hatami et al., 2015), distributed no-wait FSSP (Lin & Ying, 2016), no-wait FSSP (Shao et al., 2017), two-machine no-wait FSSP with uncertain setup times (Allahverdi & Allahverdi, 2018), cyclic two machine FSSP (Bożejko et al., 2020), probabilistic FSSP with job delay (Janaki & Mohamed Ismail, 2020),

* Corresponding author.
E-mail address: mthangaraj51@gmail.com (M. Thangaraj)

doi: 10.5267/j.jpm.2022.8.002

FSSP with corrective and preventive maintenance (Ye et al, 2020), precedence constrained FSSP with job weights (Thangaraj et al., 2022), and *m*-machine no-wait FSSP with time bounds (Allahverdi et al., 2022).

Concerning the solution methods, several techniques including heuristics and exact approaches have been devoted to the classical FSSP and its variants. Because of FSSP NP-hard nature, researchers have given much attention to the heuristic algorithms (Liang et al., 2022). These methods provide solutions depending on problem-oriented context and construction principles, which may not produce the optimal sequence but can assure the processing sequence's local optimal solution to a level. According to existing research works, the NEH heuristic algorithm proposed by Nawaz, Enscore, and Ham in 1983 is the best heuristic method of solving this problem (Taillard, 1990; Framinan et al., 2003). Researchers proposed various extensions to the NEH algorithm in light of its dominance in solving FSSP and the limitations of heuristic algorithms (Kalczynski & Kamburowski, 2008; Fernandez-Viagas & Framinan, 2014). To overcome the local optimality, metaheuristic techniques have emerged and made it possible to solve NP-hard problems more efficiently. To tackle FSSP allied models, several population evolution based techniques have been developed. To name few, Genetic algorithm (GA) (Salido et al., 2016), Hybrid GA (Semančo & Modrák, 2011), Simulated annealing (Jolai et al., 2012), Hybrid monkey search (Marichelvam et al., 2017), Multi-verse optimizer (MVO) (Wang et al., 2019), Discrete differential evolution (Ren et al., 2021) were developed for efficient solutions of FSSP and its variants.

It is observed that the assumption in most research on FSSP and its allied models is that machines are always available (Johnson, 1954; Li et al., 2019). This assumption, therefore, conflicts with the flow shop scheduling scenarios in real industrial systems since, in reality, maintenance must be performed, which has a substantial impact on a range of performance parameters such as efficiency, durability, and affordability (Ye et al., 2020). However, effective maintenance and production scheduling are two essential, strictly integrated themes in manufacturing systems that have received substantial attention in current years. On the one hand, production planning seeks to meet consumer expectations in a timely manner. Maintenance operations, on the other hand, aid in the restoration of machine reliability by lowering the rate of machine failure. Infact due to the sensitivity of machine deterioration, the machines may fail. As a result, maintenance must be considered alongside production. Typically, maintenance can be classified into two types: corrective maintenance (CM) and preventative maintenance (PM). When a machine fails abruptly, the CM includes the repair or replacement of the components. The CM is a kind of repair operation (not considered in this study) that is to be performed as soon as the machines fail. PM is a schedule of strategic operations that are performed prior to failure, to retain the system operating at the preferred level of consistency. Furthermore, various production process situations tend to exhibit job priority i.e. certain jobs should be performed in a specified sequential order. This scenario is formally referred to as an ordered precedence constraint, and it affects the best job sequence. It is used to enforce that some set of jobs has to process in the specified order all machines (Gladky et al., 2004; Cheng et al., 2017).

In view of the significance of PM and ordered precedence constraint in the flow shop scheduling, this study considers a three-machine FSSP with machine specific maintenance where each machine is provided with a specific time that when it should undergo maintenance, is considered. Here, each machine will immediately go for maintenance when the cumulative processing time exceeds the predefined maintenance schedule of the respective machine. In addition, a practical ordered precedence constraint in which some set of jobs has to process in the specified order irrespective of their processing times is also considered. The problem's goal is to establish the optimal job sequence and preventive maintenance such that the overall cost of tardiness and preventive maintenance is as minimum as possible. To tackle this problem, an efficient heuristic approach is designed that assures optimal or near optimal results. Study of this type of practical constraints not only leads to new results in scheduling theory, but also provides a flexible mathematical tool for handling technological or managerial decisions that specify certain job orders on different machines.

2. Problem Description

2.1 Variables, Indexes and Parameters

Variables, indices, and parameters used all over the article are listed in Table 1.

Table 1
Variables, Indices and Parameters

J	Set of jobs, $J = \{1, 2, 3, ..., n\}$
i	Index for jobs
n	Number of jobs
r	Number of jobs in the ordered precedence relation
$[q]$	Index for q^{th} job in a sequence, $q = \{1, 2, 3, ..., n\}$
M	Set of machines, $M = \{1, 2, 3\}$
j, k	Index for machine

m	Number of machines
g_i & t_i	Job carrying times from machine 1 to machine 2 and machine 2 to machine 3, respectively
f_i	Flow time of job i
MS_j	Time at which machine j should undergo maintenance
PM_j	Maintenance duration time on machine j
CT_j	Cumulative processing time of machine j
IT_j	Idle time of machine j
P_{ij}	Processing time of job i on machine j
$[\alpha]$	Priority mandatory starting job in the optimal job sequence
$[\lambda]$	A partial ordered job precedence sequence, $\lambda=\{[q_1],[q_2],.[q_r]\}$ where q_i is i^{th} job
β	A permutation job handling sequence, $\beta=\{[\alpha],..[\lambda],..[n]\}$
β^*	Optimal job sequence
M_j	Idle time of machine j
FT_j	Flow time of machine j

2.2 Assumptions

The present model is defined under the following assumptions:

- Each machine is able to process only one job at a time.
- Each job can only be processed by one machine at a time.
- All jobs have to perform on the predefined machine order.
- Processing times are deterministic.
- Machine specific preventive maintenance times i.e., the time when each machine shall undergo maintenance is deterministic and predefined.
- The time when each machine shall undergo maintenance is deterministic and predefined.
- The PM cannot stop a job from being processed once it has begun i.e., Preemptions are not allowed.
- All the machines are heterogeneous (independent to each other), which are available at time zero.
- All jobs are set at the beginning of the scheduling horizon
- Ordered precedence relation/ sub sequence and mandatory starting/ priority job are predefined.
- The jobs in the ordered precedence sub sequence must be performed in the specified order only.
- None of the machines fail during the job's execution, thus CM is ignored.
- Initially, all of the machines are presumed to be in good working order.
- When a machine is serviced, PM operation resets the system to its original state, thereby reverting the machine's age to zero.

2.3 Problem statement

The three-machine FSSP is defined as follows: Let a flow shop with three machines $M=\{1.2,3\}$ in series with unrestricted storage among the three machines. Let jobs $J = \{1,..,n\}$ are required to perform on the machine 1 followed by machine 2 and machine 3. The processing time of job $i(J=\{1,...n\}/i \in J)$ on machine 1 is P_{i1}, processing time on machine 2 is P_{i2} and the processing time on machine 3 is P_{i3}. The times required to carry the jobs from machine 1 to machine 2 and then from machine 2 to machine 3 be g_i and t_i, respectively. These times are known as conveyance/transportation times. Let a priority job α that ensures that the feasible job sequence always has the job α at its first position i.e. the sequence always starts with priority job α irrespective of its processing time. In addition, a partial ordered precedence job sequence, [λ], $\lambda = \{[q_1], [q_2], \ldots, [q_r]\}$ that always assumes to process in the specified order through all the machines. The time at which machine 1, machine 2 and machine 3 should undergo maintenance are deterministic and are indicated by MS_1, MS_2 and MS_3, respectively. Maintenance time for machine 1, machine 2 and machine 3 are predefined, which are represented by PM_1, PM_2 and PM_3, respectively. Each machine will immediately undergo maintenance when the cumulative processing time CT_j $(j=\{1,2,3\})$ exceeds the predefined maintenance schedule MS_j $(j=\{1,2,3\})$ of the respective machine. Here, idle times of machines are also utilized for the maintenance to optimize the overall elapsed time. At time zero, all jobs are independent of one another and available. More than one machine cannot execute the same operation at the same time, and each machine can only execute one operation at a time. The goal is to determine an optimal job sequence $\beta^* = \{[\alpha], \ldots, [\lambda], ..[n]\}$ that minimizes the make span and maximizes the machine accessibility. The layout of the stated problem is depicted in Table 2. Fig. 1 depicts a schematic representation of the present problem.

Table 2
Problem layout

Job (i)	Processing time on Machine 1 (P_{i1})	Conveyance time (g_i)	Processing time on Machine 2 (P_{i2})	Conveyance time (t_i)	Processing time on Machine 3 (P_{i3})
1	P_{11}	g_1	P_{12}	t_1	P_{13}
2	P_{21}	g_2	P_{22}	t_2	P_{23}
⋮	⋮	⋮	⋮	⋮	⋮
n	P_{n1}	g_n	P_{n2}	t_n	P_{n3}

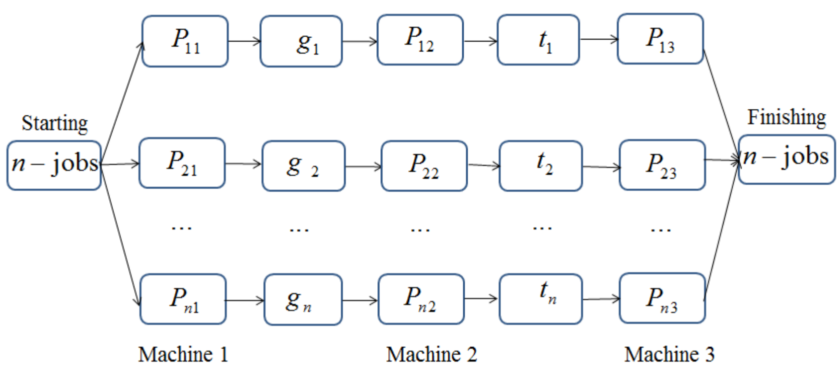

Fig. 1 Schematic representation of the present model

3. Proposed Algorithm

This section presents a simple and efficient solution methodology for determining the various performance measures for a 3-machine flow-shop scheduling problem with transportation and preventive maintenance times, and priority jobs.
The systematic proposed algorithm is described below:

Step 1: Convert the given three machine problem into two-machine FSSP by using the following formulas $S_{ij} = P_{i1} + g_i + P_{i2} + t_i$ and $Q_{ij} = g_i + P_{i2} + t_i + P_{i3}$
where S_{ij} and Q_{ij} are processing times of revised problem.

Step 2: Compute the optimal sequence β^* for the revised problem using Johnson's algorithm by ensuring the optimal sequence having priority job $[\alpha]$ in its starting position and an ordered precedence job sequence $[\lambda]$ in its appropriate place. During Johnson's algorithm process, if any job belongs from the ordered precedence subsequence job set, then we include it in the optimal sequence. Proceed further to compute the complete optimal sequence for rest of the jobs.

Step 3: Determine the total elapsed time to the given problem using the optimal sequence obtained in Step 2.

Step 4: Calculate the cumulative flow time (CT_j) for all the machines. If the cumulative flow time of the machine j exceeds its corresponding machine specified maintenance time (MS_j) of the machine j, then allow the j^{th} machine to undergo for maintenance. This process has to perform for all the machines.

Step 5: Identify the jobs, which are affected by the PM.

- If the jobs are affected by the PM, then update the initial table by adding the maintenance time to the processing times of those jobs.
- Else, retain the same processing times in the initial table.

Step 6: Using the revised initial table which is obtained in Step 5 and the optimal sequence, calculate the total elapsed time, flow time of each job, flow time of each machines and machine idle time.

4. Numerical Example

The systematic process of the proposed algorithm is demonstrated with the help of suitable numerical examples. Let us consider a three-machine flow shop scheduling problem on six jobs with conveyance times. The priority job is job [α]=[3]. The ordered precedence job sequence is [λ]={[5],[2],[4]} i.e., the specified jobs should strictly process in the order 5th job followed by 2nd job and then 4th job i.e. 5-2-4. The objective of the given problem is to minimize the total elapsed time. The processing and conveyance times (in hours), and PM schedule times (in hours) are given in Table 3 Table 4 as follows:

Table 3
Numerical instance with conveyance times

Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
1	11	4	14	2	10
2	8	3	10	4	9
3	7	2	9	3	12
4	10	5	12	2	6
5	9	2	11	3	11
6	12	3	12	4	13

Table 4
Machine specific PM times

Machines (j=1,2,3)	Time (in hours)to start Maintenance (MS_j)	Time (in hours) taking for Maintenance (PM_j)
Machine 1	25	5
Machine 2	30	3
Machine 3	35	2

Step 1: Initially, the given three-machine problem is reduced into two-machine problem using the given conditions and the resultant problem is given in Table 5.

Table 5
Reduced two-machine flow-shop problem

Job (i)	S_{ij}	Q_{ij}
1	31	30
2	25	26
3	21	26
4	29	25
5	25	27
6	31	32

Step 2: This step is devoted to finding the optimal job sequence using the Johnson's algorithm for the reduced problem. Since the priority/mandatory starting job is [α] = [3], it is obvious that job 3 has to process first irrespective of its processing time. Subsequently, the ordered precedence partial job sequence is 5-2-4, it should be employed in appropriate place in the final optimal job sequence. Now, the Johnson's algorithm can be applied to the rest of the jobs by excluding priority job and ordered precedence partial job sequence, which results the final optimal sequence for all the jobs as shown below:

3	5	2	4	6	1

Steps 3-4: The total elapsed time is 90 hours that is calculated using the above optimal sequence and the PM affected jobs are identified as Job 1 for Machine 2, Job 2 for Machine 2 and Job 4 for Machine 1 and Machine 3, which are highlighted in bold as shown in Table 6.

Table 6
Total elapsed time with the absence of PM

Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
3	0-7	2	9-18	3	21-33
5	7-16	2	18-29	3	33-44
2	16-24	3	**29-39**	4	44-53
4	**24-34**	5	39-51	2	**53-59**
6	34-46	3	51-63	4	67-80
1	46-57	4	**63-77**	2	80-90

Step 5: The revised processing times after adding the maintenance times to those PM affected jobs is given in Table 7. However, Job 4 for Machine 3 is affected by PM time, its respective processing time remains the same. This is due to the fact that Machine 3 will be idle for 8 hours between job 4 and job 6. During that period, the Machine 3 is allowed to get maintenance (maintenance time is 2 hours), this strategy results in minimizing the total elapsed time.

Table 7
Effect of original processing times due to PM times

Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
1	11	4	**17**	2	10
2	8	3	**13**	4	9
3	7	2	9	3	12
4	**15**	5	12	2	6
5	9	2	11	3	11
6	12	3	12	4	13

Step 6: The total elapsed time, flow time of each job and idle time of the machines are determined using the revised job processing times, which are reported in Tables 8-9. Finally, the end solution is represented through Gantt chart shown in Fig. 2.

Table 8
Total elapsed time due to effect of PM times

Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
3	0-7	2	9-18	3	21-33
5	7-16	2	18-29	3	33-44
2	16-24	3	29-42	4	46-55
4	24-39	5	44-56	2	58-64
6	39-51	3	56-68	4	72-85
1	51-62	4	68-85	2	87-97

Table 9
Summary of results

Ordered precedence partial job sequence	Priority job	Optimal Sequence	PM affected jobs	Overall flow time of each job (in hrs)	Idle time of each machine (in hrs)	Overall flow time of each Machines (in hrs)	Total elapsed time due to PM (in hrs)
5-2-4	3	3-5-2-4-6-1	1,2 & 4	$f_3 = 33$ $f_5 = 36$ $f_2 = 37$ $f_4 = 40$ $f_6 = 44$ $f_1 = 44$	$M_1 = 0$ $M_2 = 11$ $M_3 = 36$	$M_1 = 62$ $M_2 = 74$ $M_3 = 61$	97

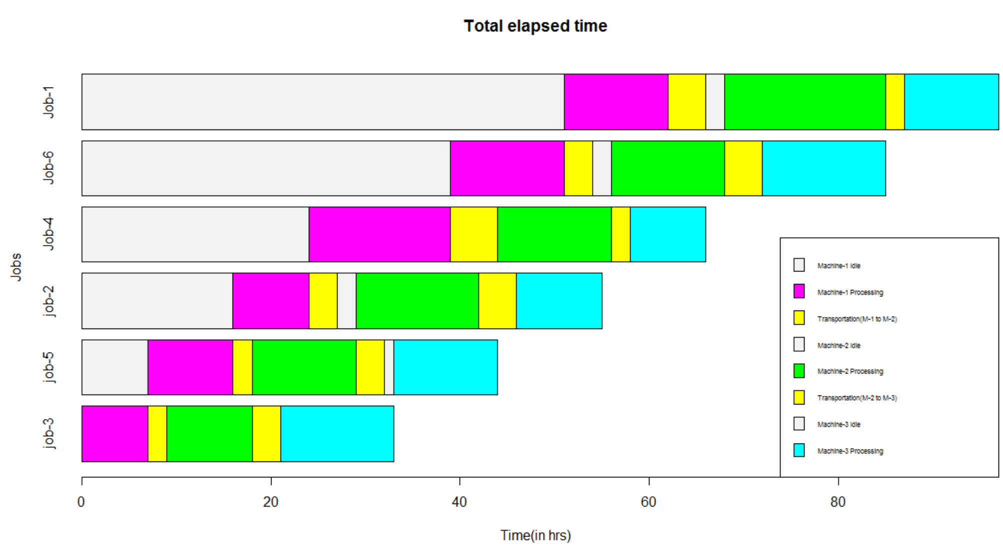

Fig. 2. Gantt chart for the solution

5. Computational Results

The computation results are presented in this section. The developed algorithm was programmed and tested in MATLAB 2021a, and all experiments were performed on a PC with an Intel Core i5 processor running at 2.10 GHz, 4 GB of RAM, and Microsoft Windows 2010 OS. Since there are no previous research on the current model, this study has not performed any comparative tests to evaluate the algorithm's performance in terms of solution quality. However, to test the algorithm's performance in computational time aspect, a set of five numerical random test instances (given in the annex) has been generated, which are indicated by IN1, IN2, IN3, IN4, and IN5, respectively. From these five instances, a set of 12 test cases has been generated with different combinations of the parameters. For each instance, PM times and precedence constraints are changed. All these test instances were tested on the proposed algorithm and results are reported in Table 10. The reported results shall be used for future comparative studies.

Table 10
Computational results

Instance Name & its size	Priority job & Ordered precedence jobs	MS_j	PM_j	Optimal Sequence	PM affected jobs	Overall flow time of each job	Idle time of each machine	Overall flow time of each Machines	Total elapsed time due to PM
IN1 (3×8)	5 & 4-6-2	100, 80, 60	5, 3, 5	5-3-7-8-1-4-6-2	6,7 & 8	$f_5 = 69$ $f_3 = 86$ $f_7 = 83$ $f_8 = 76$ $f_1 = 69$ $f_4 = 65$ $f_6 = 68$ $f_2 = 64$	$M_1 = 0$ $M_2 = 56$ $M_3 = 65$	$FT_1 = 185$ $FT_2 = 151$ $FT_3 = 179$	244
IN1 (3×8)	2&1-3-5	80, 90, 100	10, 5, 10	2-4-7-8-1-3-5-6	1,6 & 8	$f_2 = 64$ $f_4 = 65$ $f_7 = 80$ $f_8 = 80$ $f_1 = 69$ $f_3 = 86$ $f_5 = 69$ $f_6 = 73$	$M_1 = 0$ $M_2 = 66$ $M_3 = 67$	$FT_1 = 200$ $FT_2 = 150$ $FT_3 = 171$	238
IN2 (3×5)	2 &3-5	2000, 2500, 5000	100 , 150 , 200	2-3-5-1-4	1,3 & 5	$f_2 = 4425$ $f_3 = 4758$ $f_5 = 4291$ $f_1 = 4338$ $f_4 = 4349$	$M_1 = 0$ $M_2 = 3280$ $M_3 = 1911$	$FT_1 = 6325$ $FT_2 = 3930$ $FT_3 = 11569$	13480
IN2 (3×5)	1&2-4	3000, 4000, 5000	50, 75, 100	1-2-4-5-3	4	$f_1 = 4238$ $f_2 = 4425$ $f_4 = 4499$ $f_5 = 4091$ $f_3 = 4658$	$M_1 = 0$ $M_2 = 3047$ $M_3 = 2084$	$FT_1 = 6175$ $FT_2 = 3930$ $FT_3 = 11469$	13553

Table 10
Computational results (Continued)

Instance Name & its size	Priority job & Ordered precedence jobs	MS_j	PM_j	Optimal Sequence	PM affected jobs	Overall flow time of each job	Idle time of each machine	Overall flow time of each Machines	Total elapsed time due to PM
IN3 (3×15)	10 & 4-8-6-5-12	200, 300, 350	10, 12, 14	10-7-4-8-6-5-12-14-1-15-3-11-9-13-2	1,12 & 13	$f_{10}=153$ $f_7=94$ $f_4=149$ $f_8=145$ $f_6=119$ $f_5=144$ $f_{12}=109$ $f_{14}=105$ $f_1=148$ $f_{15}=136$ $f_3=146$ $f_{11}=151$ $f_9=124$ $f_{13}=129$ $f_2=104$	$M_1=0$ $M_2=51$ $M_3=171$	$FT_1=510$ $FT_2=682$ $FT_3=612$	783
IN3 (3×15)	3 & 2-9-14-11	200, 300, 350	10, 12, 14	3-7-12-6-5-1-15-4-8-10-13-2-9-14-11	4,14 & 11	$f_3=146$ $f_7=94$ $f_{12}=87$ $f_6=119$ $f_5=144$ $f_1=134$ $f_{15}=136$ $f_4=185$ $f_8=145$ $f_{10}=153$ $f_{13}=105$ $f_2=104$ $f_9=124$ $f_{14}=115$ $f_{11}=163$	$M_1=0$ $M_2=27$ $M_3=148$	$FT_1=510$ $FT_2=682$ $FT_3=612$	760
IN3 (3×15)	10 & 4-8-6-5-12	250, 275, 300	20, 25, 30	10-7-4-8-6-5-12-14-1-15-3-11-9-13-2	1,2,5,9 & 14	$f_{10}=153$ $f_7=94$ $f_4=149$ $f_8=145$ $f_6=119$ $f_5=169$ $f_{12}=87$ $f_{14}=135$ $f_1=154$ $f_{15}=136$ $f_3=146$ $f_{11}=151$ $f_9=149$ $f_{13}=105$ $f_2=104$	$M_1=0$ $M_2=51$ $M_3=181$	$FT_1=510$ $FT_2=708$ $FT_3=628$	809

Table 10
Computational results (Continued)

Instance Name & its size	Priority job & Ordered precedence jobs	MS_j	PM_j	Optimal Sequence	PM affected jobs	Overall flow time of each job	Idle time of each machine	Overall flow time of each Machines	Total elapsed time due to PM
IN4 (3×7)	2 & 3-6-4	200, 220, 250	12, 15, 18	2-7-5-3-6-4-1	3,4 & 7	$f_2=422$ $f_7=600$ $f_5=496$ $f_3=438$ $f_6=529$ $f_4=508$ $f_1=415$	$M_1=0$ $M_2=140$ $M_3=471$	$FT_1=1116$ $FT_2=1169$ $FT_3=996$	1467
IN4 (3×7)	1 & 2-5-4	250, 300, 350	20, 25, 30	1-2-5-4-7-6-3	2,3,4,5,6 & 7	$f_1=415$ $f_2=442$ $f_5=542$ $f_4=483$ $f_7=597$ $f_6=549$ $f_3=466$	$M_1=0$ $M_2=178$ $M_3=507$	$FT_1=1140$ $FT_2=1199$ $FT_3=1028$	1535
IN5 (3×10)	10 & 2-5-6-3	40, 45, 50	15, 20, 25	10-7-2-5-6-3-9-1-4-8	3,4,5 & 9	$f_{10}=44$ $f_7=29$ $f_2=37$ $f_5=79$ $f_6=41$ $f_3=55$ $f_9=62$ $f_1=44$ $f_4=75$ $f_8=30$	$M_1=0$ $M_2=62$ $M_3=37$	$FT_1=153$ $FT_2=102$ $FT_3=172$	209
IN5 (3×10)	4 & 1-2-3	50, 60, 70	10, 15, 20	4-7-1-2-3-5-6-10-9-8	3,5 & 8	$f_4=50$ $f_7=29$ $f_1=44$ $f_2=37$ $f_3=54$ $f_5=59$ $f_6=54$ $f_{10}=44$ $f_9=47$ $f_8=40$	$M_1=0$ $M_2=49$ $M_3=41$	$FT_1=143$ $FT_2=104$ $FT_3=142$	183
IN5 (3×10)	6 & 3-2-4	45, 55, 65	20, 25, 30	6-7-3-2-4-5-10-9-1-8	2,4,5 & 9	$f_6=41$ $f_7=29$ $f_3=44$ $f_2=57$ $f_4=50$ $f_5=69$ $f_{10}=63$ $f_9=67$ $f_1=44$ $f_8=30$	$M_1=0$ $M_2=69$ $M_3=58$	$FT_1=163$ $FT_2=110$ $FT_3=152$	210

6. Conclusions

The two most important components of agile manufacturing are effective maintenance and production planning. Generally, machines may fail due to lack of proper maintenance, which leads to uncertain breakdowns in the production process. Therefore, maintenance must be considered in conjunction with the production process. Most existing preventive maintenance FSSP related models are studied with a specific PM interval, which is rigid and may result in poor performance. To tackle this issue, this study is aimed at a practical three-machine jobs flow shop-scheduling problem (FSSP) in which machine specific preventative maintenance is taken into account, with each machine having its own deterministic maintenance plan. A practical ordered precedence constraint is also examined, in which a set of jobs must process in the specified order, regardless of their processing times. The problem's objective is to determine the optimal job sequence and PM such that the overall cost of tardiness and PM is minimized. To obtain optimal job sequence, an efficient heuristic approach is designed that results in total cost savings. A comparative study is not performed due to no existing studies on the current problem. However, computational experiments are carried out on some test instances and results are reported. The reported results may be useful for future studies.

References

Allahverdi, A., & Allahverdi, M. (2018). Two-machine no-wait flowshop scheduling problem with uncertain setup times to minimize maximum lateness. *Computational and Applied Mathematics*, *37*(5), 6774-6794.

Allahverdi, A., & Al-Anzi, F. S. (2006). A branch-and-bound algorithm for three-machine flowshop scheduling problem to minimize total completion time with separate setup times. *European Journal of Operational Research*, *169*(3), 767-780.

Allahverdi, A., Aydilek, H., & Aydilek, A. (2022). An algorithm for a no-wait flowshop scheduling problem for minimizing total tardiness with a constraint on total completion time. *International Journal of Industrial Engineering Computations*, *13*(1), 43-50.

Baskar, A., & Xavior, A. (2012). A Simple Model to Optimize General Flow Shop Scheduling Problems with Known Break down Time and Weights of Jobs. *Procedia Engineering*, *38*(1), 191-196.

Baskar, A., & Xavior, M. A. (2014). Optimization of makespan in job and machine priority environment. *Procedia Engineering*, *97*, 22-28.

Bożejko, W., Smutnicki, C., Uchroński, M., & Wodecki, M. (2020). Cyclic Two Machine Flow Shop with Disjoint Sequence-Dependent Setups. In *Modelling and Performance Analysis of Cyclic Systems* (pp. 31-47). Springer, Cham.

Cheng, S. R., Yin, Y., Wen, C. H., Lin, W. C., Wu, C. C., & Liu, J. (2017). A two-machine flowshop scheduling problem with precedence constraint on two jobs. *Soft Computing*, *21*(8), 2091-2103.

Fernandez-Viagas, V., & Framinan, J. M. (2014). On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. *Computers & Operations Research*, *45*, 60-67.

Framinan, J. M., Leisten, R., & Rajendran, C. (2003). Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. *International Journal of Production Research*, *41*(1), 121-148.

Gao, J., & Chen, R. (2011). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem." *International Journal of Computational Intelligence Systems, 4*(4), 497-508.

Gladky, A. A., Shafransky, Y. M., & Strusevich, V. A. (2004). Flow shop scheduling problems under machine–dependent precedence constraints. *Journal of combinatorial optimization*, *8*(1), 13-28.

Hatami, S., Ruiz, R., & Andrés-Romano, C. (2015). Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. *International Journal of Production Economics*, *169*, 76-88.

Janaki, E., & Mohamed Ismail, A. (2020). Flow Shop Scheduling in Which Processing Time Connected with Probabilities and Job Delay Due to Maintenance for M* N Machine. In *Information and Communication Technology for Sustainable Development* (pp. 651-657). Springer, Singapore.

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, *1*(1), 61-68.

Jolai, F., Rabiee, M., & Asefi, H. (2012). A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *International Journal of Production Research*, *50*(24), 7447-7466.

Kalczynski, P. J., & Kamburowski, J. (2008). An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers & Operations Research*, *35*(9), 3001-3008.

Lee, J. Y., & Kim, Y. D. (2017). Minimizing total tardiness in a two-machine flowshop scheduling problem with availability constraint on the first machine. *Computers & Industrial Engineering*, *114*, 22-30.

Li, W., Nault, B. R., & Ye, H. (2019). Trade-off balancing in scheduling for flow shop production and perioperative processes. *European Journal of Operational Research*, *273*(3), 817-830.

Liang, Z., Zhong, P., Liu, M., Zhang, C., & Zhang, Z. (2022). A computational efficient optimization of flow shop scheduling problems. *Scientific Reports*, *12*(1), 1-16.

Lin, S. W., & Ying, K. C. (2016). Minimizing makespan for solving the distributed no-wait flowshop scheduling problem. *Computers & Industrial Engineering*, *99*, 202-209.

Marichelvam, M. K., Tosun, Ö., & Geetha, M. (2017). Hybrid monkey search algorithm for flow shop scheduling problem under makespan and total flow time. *Applied Soft Computing*, *55*, 82-92.

Ren, T., Wang, X., Liu, T., Wu, C. C., Bai, D., Lin, L., & Guo, M. (2021). Exact and metaheuristic algorithms for flow-shop scheduling problems with release dates. *Engineering Optimization*, 1-17.

Ruiz, R., García-Díaz, J. C., & Maroto, C. (2007). Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers & Operations Research*, *34*(11), 3314-3330.

Salido, M. A., Escamilla, J., Giret, A., & Barber, F. (2016). A genetic algorithm for energy-efficiency in job-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *85*(5), 1303-1314.

Semančo, P., & Modrák, V. (2011, October). Hybrid GA-based improvement heuristic with makespan criterion for flow-shop scheduling problems. In *International Conference on ENTERprise Information Systems* (pp. 11-18). Springer, Berlin, Heidelberg.

Shao, W., Pi, D., & Shao, Z. (2017). Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms. *Knowledge-Based Systems*, *137*, 163-181.

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European journal of Operational research*, *47*(1), 65-74.

Thangaraj, M., Kumar, T., & Nandan, K. (2022). A precedence constrained flow shop scheduling problem with transportation time, breakdown times, and weighted jobs. *Journal of Project Management*, *7*(4), 229-240.

Wang, H., Huang, M., & Wang, J. (2019). An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs. *Journal of Intelligent Manufacturing*, *30*(7), 2733-2742.

Ye, H., Wang, X., & Liu, K. (2020). Adaptive preventive maintenance for flow shop scheduling with resumable processing. *IEEE Transactions on Automation Science and Engineering*, *18*(1), 106-113.

Appendix

Name	Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
IN1	1	24	4	13	6	22
	2	12	7	15	5	25
	3	31	2	22	3	28
	4	17	6	17	4	21
	5	19	5	25	2	18
	6	25	3	13	5	17
	7	28	2	24	2	24
	8	24	5	21	4	16

Name	Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
IN2	1	1056	52	952	24	2154
	2	1115	45	716	35	2514
	3	1419	48	754	20	2417
	4	1214	35	850	15	2235
	5	1321	38	658	25	2049

Name	Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
IN3	1	35	5	45	7	42
	2	42	4	20	4	34
	3	21	6	62	5	52
	4	43	5	53	3	45
	5	18	4	52	5	65
	6	25	6	42	4	42
	7	26	3	24	6	35
	8	35	5	68	5	32
	9	41	6	48	4	25
	10	44	7	54	7	41
	11	32	4	64	6	45
	12	24	5	25	5	28
	13	35	6	28	4	32
	14	27	4	32	6	36
	15	42	4	41	5	44

	Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
IN4	1	158	10	124	8	115
	2	112	12	168	10	120
	3	141	9	133	6	122
	4	150	12	152	9	140
	5	165	8	171	6	146
	6	180	7	182	8	152
	7	174	10	194	12	165

	Job (i)	P_{i1}	g_i	P_{i2}	t_i	P_{i3}
IN5	1	10	4	15	5	10
	2	12	2	5	4	14
	3	18	5	6	3	12
	4	17	3	12	2	16
	5	9	4	7	4	15
	6	12	2	10	5	12
	7	8	3	8	2	8
	8	10	4	6	3	7
	9	15	5	10	2	15
	10	12	2	12	5	13