Journal of Research Administration
Society of Research Administrators International

JOB SHOP SCHEDULING WITH NON-BOTTLENECK MACHINESCONSIDERING THE MAINTENANCE ASPECTS

Gorle Mangamma[1], M. SrinivasaRao[2], V.V.S. Kesavarao[3]
[1]Part-time Ph.D. Scholar in Department of Mechanical Engineering, College of Engineering(A), Andhra University, V isakhapatnam-3.

[2]DGM (Maintenance), Department of MMSM, Visakhapatnam Steel Plant.
[3]Professor, Department of Mechanical Engineering, College of Engineering(A), Andhra University, Visakhapatnam.

Abstract
Optimising the scheduling of current machines is related to machine scheduling. It is a common premise that machines are available whenever need arises. However, it is not true always as a machine may become nonfunctional at certain point oftimes. The problem of the job shop in a dynamic environment is attempted to be solved. Job shop maintenance issues are challenging optimisation issues. To reduce the likelihood of a machine breaking down, predictive maintenance is a viable option. The three objectives are to prioritise the volume of handled jobs, decrease the completion time, and consequently decrease the finishing times of the non-bottleneck machine in unrelated preventive maintenance scheduling (upms). To find roughly workable solutions, a hybrid tabu search (TS) method and mixed-integer programming (MILP) model are developed According to computational findings, the hybrid TS approach enables quick acquisition of the most processed jobs (average 8 s). The first two goals have been accomplished by the MILP model. All three objectives are satisfactorily attained by the hybrid TS algorithm. The hybrid TS algorithm's third phase also demonstrates its efficiency in increasing equipment utilisation.
Keywords: scheduling, breakdown, machine, random

1. Introduction

Production scheduling is a method of assigning a specific sequence of tasks to a predetermined number of machines while maintaining a target performance level. All jobs are processed through a machine as per agreed order. There are specific processes for each job I that have set processing timeframes. One machine j is able to perform Pij if it is known in advance. All jobs may be passed through different routes. Allocating jobs one by one on each designated machine. In reality, production systems are less precise and certain. In reality, a lot of things happen, including new jobs, arrival machine failures, rearranged deadlines, cancelled orders, and the appearance of an emergency situation.

The maintenance time of machines are to be considered with the job scheduling to enhance improve the manufacturing unit performance. Preventive maintenance tasks are designed for each machine to reduce breakdowns. As a result, it turns into a limitation for the production scheduling issue. The dynamic job shop scheduling challenge brought on by unforeseen machine breakdowns can be resolved by implementing an effective preventative maintenance program.

The section 2 literature review is presented. In Section 3, scheduling methods are introduced. Section 4 describes computational results and findings, while Section 5 discussions. Section 6 narrates a conclusion and Section 7 contains future scope of study.

2. Literature Review

Allocating resources to execute tasks over time in accordance with certain criteria is the subject of job shop scheduling. Dynamic job shop deals with change in criteria dynamically such as machine breakdowns and it is which is a NP-hard combinatorial optimization problem.

Preventive maintenance and single machine production scheduling were covered by Chang HC., et al. [1]. They regarded the best possible order for jobs including the best preventive maintenance scheduling using a Genetic Algorithm (GA) technique to be reducing the weighted predicted finishing time of the jobs. Paprocka [2] suggested using high level prediction to incorporate manufacturing equipment degradation into scheduling. In order to study work shop scheduling, which comprised sequence-dependent preparation durations and preventative maintenance for a single target, MH Abd Elrahman Elgendy [3] used Taillard's instances. The maintenance is to be performed to prevent sudden breakdown and jobs are also to be performed on the same machine. In order to study work shop scheduling, which comprised sequence-dependent preparation durations and preventative maintenance for a single target, MH Abd Elrahman Elgendy [3] used Taillard's instances. Preemption is authorised to quicken task scheduling and reduce Cmax (makespan). is taken into account with the following presumptions. Entire machines planned for process are in working condition at the start of schedule.

- Any One machine randomly assumed in failure condition is permitted.
- Precedence relationship. Exists in sequence of operation
- The processing time of operations known in advance.
- One operation on one machine at any single instant.
- The preemption is permitted.
- Only after an existing work has been completed will new jobs enter the scheduling process. The following are the problem's presumptions.
- One machine at most may be used to process each job; preemption is not permitted. Processing durations vary. On various machines, processing times for a work can differ.
- Machine inactivity is permitted.
- The machines' malfunction and failure are taken into consideration one job and taken by each machine at once.
- A dispatch time exists for every job. Each machine has time available.
- Every match between a machine and a job has an expiration date. Every machine can only complete certain tasks.

The instruction of DJSSP is made in Ben Ali [4]. The DJSSP's goal is to distribute new jobs as soon as they enter the system while it is in operation. The Gentic algorithm is applied to tackle newtasks due to its superiority which has been thoroughly investigated by Graves G., et al. [5], Lee W-C. [6], Brandolese M., et al. [7], Ben Ali [4].

Regarding simultaneous job shops scheduling and job shops with preventative maintenance, less material is available. Considering the aspects of preventative maintenance in a dynamic setting. The study is carried out two kinds of machine environments. The study's primary goal is to schedule work in a dynamic setting with the possibility of unexpected machine failure by minimising the time taken for the last job to be completed. The other one of the study is considered scheduling the jobs on Unrelated Machines environment with Inclusion Preventive Maintenance (upms).

3. Methods

3.1 Problem narration for related machines

A group M = l,...,m of m machines must be scheduled to handle a set I = l,...,n of n independent jobs of various sizes. Each of the m machines must process each of the ni ordered operations Ori, 1,..., Ori.ni that make up a task. Let Or = 0, 1, o, and o + 1 represent the collection of all operations that need to be scheduled. Operation k in set Or has processing time P_{rij} that is fixed. One process can only be done by each machine m at once, and once processing begins on a given machine, it must continue there without interruption until it is stopped. The operation that came before kO should be p_k. Operation-related constraints fall into two categories. Only idle machines on which kO is processed can be scheduled. Second, due to the requirement of precedence restrictions, each operation kO must be scheduled after its predecessor operation p_k is complete.

The objective function equation (1) minimizes the make span

$$\text{Mine } C_{max} = F_{o+1} \quad (1)$$

Subject to:

$$F_{rpk} + P_{rij} \square\square F_{rk};\ k = 1,\ldots,o+1, \quad (2)$$

Enforces the order of activities inside a single job, while placing constraints on

$$F_{ri1} - F_{ri2} \square\square P_{rij} \text{ or } F_{ri} - F_{ri2} \square P;\ (i1,i2) \square O_k \quad (3)$$

Equation (3) imposes the precedence relations between operation on

$$rF_i \square 0;\ k = 1,\ldots,o+1 \quad (4)$$

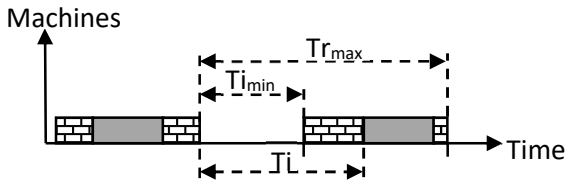

Figure 1 Period of a preventive maintenance (upkeeing p activity)

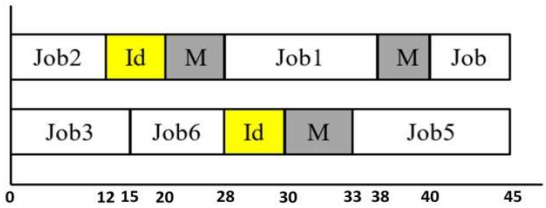

Figure 2 preventive maintenance with scheduling

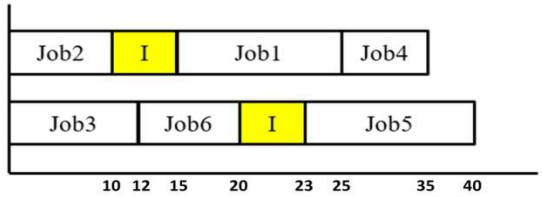

Figure 3 Part– Including preventive maintenance with scheduling

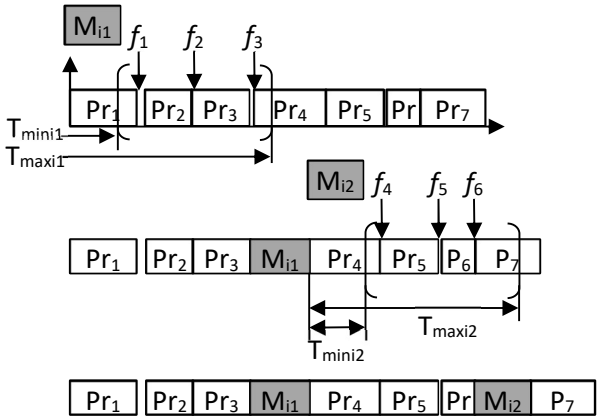

Figure 4 Possible cases of inserted operation in PM model

Table 1 Parameters

Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	PM	
1	1	2	1	2	2	15	20

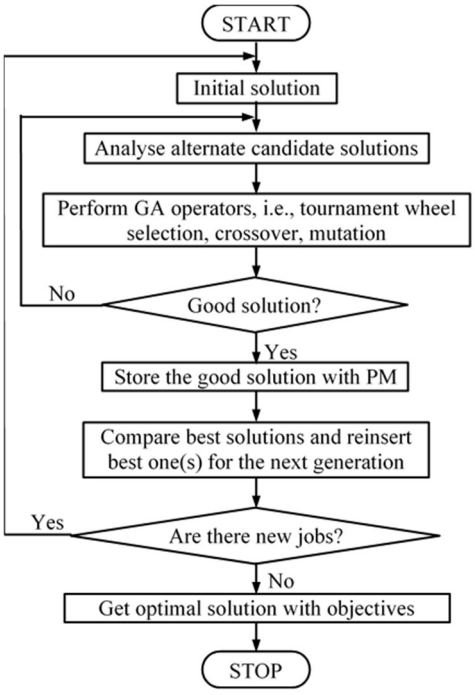

Figure 5 GA flow chart

Scheduling and Preventive Maintenance

Studying the interrelationship activity between machine availability risk and cost in broadly. There are other works using machine learning approaches in Teoh YK., et al. [17] and Yang Z., et al. [18]. Jin Y-L [15] presented the single-machine problem in Graves G. [12] with weighted finish time as the goal. Reza et al. [16] etc. The management, scheduling, and preventative maintenance of jobs on a machine were all taken into consideration by the researchers.

Integrated Strategy

The task of maintaining or restoring an equipment or resource to a particular usable state is referred to as maintenance. Corrective maintenance (CorM) and preventive maintenance are two types of maintenance(PreM). The CorM is performed after failure occurs. The rePM is carried out for a fixed time at regular intervals before failure takes place.

The operating time of PM task j on machine i is prij, they are fixed, nonnegativewithin interval [Tmin, Tmax]. A PM operation shouldideally be planned within the range [Tmin, Tmax], as shown in Fig. 1. Additionally, if some PM operations in Fig. 2 are scheduled after Tmax. The effect would be that the machine wouldn't work and that upkeep would cost more.

Let:

- Mrij: the PM task j on machine i.
- Trij: time Mirj.
- Trminij: earliest time r of Mrij.
- Trmaxij: latest time r of Mrij.
- Pij: time of PM task Mij. ot changing, fixed.

The makespan minimization is the sequential strategy's primary goal function. An overall objective goal that explains for both production and maintenance criteria is what is being optimised. A scheduling answer to address the DJSSP with the help of GA approach is proposed. every chromosome is considered as a configuration and a solution. It comprises of two components; one is Production scheduling and the other one is Preventive maintenance. Each task is allocated to a certain chromosome, which will be inspected and changed to reduce the make span (completion time ofthe last task on scheduling system). Table 1 gives a brief example with 6 jobs scheduled on 2 machines with preventative maintenance values for each machine (M1, M2) of 20 and 15 (LB, UB). LB is the maximum between the minimum of the current execution time and the beginning of the following task, as opposed to UB, which is the minimum between the last job's execution time and the minimum execution time. An integrated genetic algorithm for the DJSSP. All details are given in Table 1. Thefollowing is a description of the common GA operators: Crossover. It is used on the industrial process in order to increase output.

Mutation

It is used on the production process in order to diversify. Only the production sequence is subject to the swapping moves; the maintenance values remain unaltered. Mutation swapping is used.

In Figure 5, it is depicted that the process starts with a random initial solution as part of a preventative maintenance (PM) strategy.

Only the production sequence is subject to the swapping moves; the maintenance values remain unaltered. By switching the sequence in which the production jobs are completed, this tactic aims to produce new people. We use random genetic operators during each iteration to diversity the resulting population.

The suggested strategy takes into account the quantity of jobs, machines, and unexpectedly arriving work while taking scheduling time into consideration immediately and retaining the preventative maintenance components.

As shown in Table 2, jobs are represented by n, machines are represented by m, and unexpectedly created new jobs are represented by n'. The indicator of the performance evaluation of the proposed system is minimization of makespan, which is represented by Cmax in these cases.

The. Commonly applied dispatching routines is as follow:

1) The quickest operating time
2) The longest operating time rule.

Table-2: Performance for a jobs with machines

Instance size	**Metaheuristics**		**Dispatching rules**		
n × m ×n□	**TS**	**GA + PM**	**SPT**	**LPT**	**MWKR**
(10 × 3 × 5)	1334	1228	2411	2234	1984
(20 × 3 × 10)	1446	1354	2591	2412	2256
(30 × 5 × 5)	1532	1428	2694	2518	2437

The suggested method-based GA strategy beats the TS meta heuristic and all employed dispatching rules in terms of make span minimization without incorporating the notion of preventative maintenance, according to Table 2. The proposed method's makespan is shown to be the shortest – or approximately theshortest – among these approaches. The DJSSP can besolved using Table IV. Optimizing the performance measure make span for various arrival rates for jobs. GA with preventive maintenance. Especially, for 200 × 45 × 70, The shortest of all the values collected, the Cmax is equal to 4598. We can draw the conclusion that GA for DJSSP is effective in resolving the problem. The scheduling process is displayed in Fig. 5.

4. Methodology Adopted for Unrelated Machines with Inclusion pf Preventive Maintenance Jobs

Depending on the batch preparation time, setup time, batch (family) or non-batch preparation time, and independent or dependent setup time, the DSPs with preparation times can be categorised into four groups.as per (Allahverdi et al., [8]). The scheduling of independent parallel computers is taken into account for sequence-dependent preparation, which is dependent on job dispatch instant and lapsed instance of permitting a work to be finished on a given piece of equipment, referred to as R|rj, eij, and STsd|Cmax.. The first of the three phase aims is to increase the quantity of jobs

handled. The second is to shorten the amount of time (make span) with inclusion of preventive maintenance time, and thethird is to shorten the non-critical machines' maximumcompletion times.

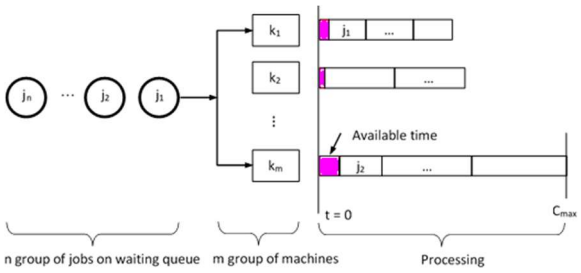

Figure 6 The scheduling process

A three-phase optimization technique is used to resolve the issue of disconnected parallel machines with sequence-dependent setup durations. The MIP model's first two objectives are to maximize handled jobs and shorten the make span. Additionally, final goal of reducing the completion times of the machines that are not bottlenecks,

4.1 The R|rj, eij, sTsd|Cmax terms

SETS

ir	Machine index I = 1,..., m), total no. of machines.
jr,kr	Job index (j = 1,..., n), is the total no of jobs. Theletters "j" and "k" refer to 2 neighboring Jj and Jk. Die job is represented by index J in this study.
Mrj	The collection of devices, M, that can perform Jrj, j J
Jri	List of jobs, J, which are handled by device Mri is as follows: j….. I Mj, j J.

PARAMETERS

prij	duration of Jj's processing on the device Mi, I M, j, and Ji.
Srijk	The machine setup time for changing from Jj to Jk is Mi I M, j, k Ji, j k.
r	The transition period from one job to another on the machine Mi, I M, j, and Ji
Srjk	The transition time from job Jj to jobs Jk, j, k, j, k, and k
Ari	The machine's current availability Mi, I □ M
Rrj	The job's release date Jj, j □ J
eirj	The time limit for processing work Jj on the machine Mi, I M, and Ji has passed.
K	a huge number.

MPrij = Maintenance processing time

Phase-1 Model:

$$\sum_{j\in J}\sum_{i\in M_j} y_{ij} \quad (5)$$

Subject to:

$$\sum_{i\in M_j} y_{ij} \le 1 \quad \text{for all } j \in J \quad (6)$$

$$x_{ik}^0 + \sum_{j\in J_i, j\neq k} x_{ijk} = y_{ik} \quad \forall i\in M,\ k \in J_i \quad (7)$$

$$\sum_{k\in J_i, k\neq j} x_{ijk} \le y_{ij} \quad \forall i\in M,\ k \in J_i \quad (8)$$

$$\sum_{j\in J_i} x_{ij}^0 \le 1 \quad \forall i\in M \quad (9)$$

$$ci \quad \text{for all } j \in \text{jobs } J \quad (10)$$

$$b_j - \max(R_j, A_i) + L\left(1 - x_{ij}^0\right) \ge 0$$

$$\text{for all i set of M, } j \in J_i \quad (11)$$

$$c_j - b_j + L\left(1 - x_{ij}^0\right) \ge P_{ij} + S_{ij}^0 \quad \forall i\in M,\ k \in J_i \quad (12)$$

$$b_k - c_j + L^*(1 - x_{ijk}) \ge o$$

$$\text{for all } i\in M,\ j, k \in J_i,\ j \neq k \quad (13)$$

$$c_k - b_k + L^*(1 - x_{ijk}) \ge P_{ik} + S_{ijk}$$

$$\text{for all } \forall i\in M,\ j, k \in J_i,\ j \neq k \quad (14)$$

$$b_j - e_{ij} - L^*(1 - y_{ij}) \le 0 \text{ for all } \forall i\in M,\ j \in J_i \quad (15)$$

$$y_{ij} \in \{1, 0\} \quad \text{for all } \forall i\in M,\ j \in J_i \quad (16)$$

$$x_j^0 \in \{o, 1\} \quad \forall i\in M,\ j \in J_i \quad (17)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i\in M,\ j, k \in J_i,\ j \neq k \quad (18)$$

$$b_j \ge o \quad \text{for all } \forall j \in J \quad (19)$$

$$c_j \ge o \quad \text{for all } \forall j \in J \quad (20)$$

While cj and bj are non-negative, binary variables with a threshold of 0.1, the decision variables yij, xijk, and bj are positive, continuous variables. The goal function in equation (5) maximizes the quantity of jobs processed in phase 1. Equation (6) states that just one computer should handle a given job and that no other machines should handle it. Equations (7)-(8) stated that when both yij and yik equal 1, Jk must follow Jj the machine Mi instantly. Equation (9) states that every machine can handle a single job for a specific time as the initial job. Jj's beginning preparation time shouldn't be earlier than its dispatch time, according to equation (10). Equation (11), if Jj is the first task handled on machine Mi, ensures that Jj's beginning setup time is longer than its dispatch time and the machine's accessible time. The time difference between Jj's beginning setup and finish time should be more than the product of those durations, according to equations (12) and (14). Equation (13), infers the successor's starting preparation time must be not earlier than the predecessor's finish time. Equation (15) makes sure that Jj can start processing on machine Mi

before its processing time has run out. The non-negativity constraints were specified in equations (16) through (20).

Phase 2 Model:

$$\text{Min} C_{max} \quad (21)$$

Subject to: Constraints (2)-(16) in phase 1.

$$C_{max} \geq C_j \qquad \forall j \in J \quad (22)$$

$$\sum_{j \in J} \sum_{i \in M_j} y_{ij} = \text{Phase 1 Model (objective value)} \quad (23)$$

$$ck - bk + L(1 - x_{ijk}) \geq Pik + S_{ijk} + mp_{rij} \quad 24.1$$

$$C_{max} \geq 0 \quad (24)$$

Step 2's main goal is to shorten the make span (equation (21)). Equation (22) satisfy that Cmax: maximum time for doing all of the tasks.

Constraint (23), who reported the number of finished jobs, based his estimate on the objective value of the previous phase 1

4.2 Combining of objectives

Even if a slack form is ultimately chosen, the total of setup time and process time should be equal (see limits (12) and (14)) between the start of setup and the job's completion time. By dramatically reducing the no. of binary variables in the third set, Mj. and Ji, we can improve the model's numerical performance.

When two objectives are combined into one objective function, it can be written as a one-phase model. There are two strategies being thought about:

1. $\text{Minimize} \left(\sum_{j \in J} 1 - \sum_{j \in J} \sum_{i \in M_i} y_{ij} \right) C_{max}$
2. $\text{Minimize} -W \sum_{j \in J} \sum_{i \in M_j} y_{ij} + C_{max}$

The model becomes a MLP thanks to the initial objective function. I (Bradley et al., 3). In order to express the proportionate relevance of the number of completed jobs, a weight factor W is added to the second equation. This is due to the fact that the number of processed jobs is typically less than 100 and the maximum completion time is more than 10,000. Even though the W is calculated empirically, processing volume matters more than completion speed. To make the model simpler, a slack form is included for the period of time between the start of the job and when it is finished (see limitations (12) and (14)).

The constraints are stated below:

Constraints:

$$ci_k - ib_k + L \geq Pi_{ik} + S_{ijk},$$

$$ck - b_{ik} = Pi_{ik} + S_{ijk}.$$

Slack form:

$$^{i}c_k - b_k + L(1 - x_{ijk}) \geq Pi_{ik} + S_{ijk}.$$

i.

The introduction of the sets Mj and Ji aims to boost numerical effectiveness. Only a few machines, as opposed to all of them, can process a certain Jj. Similarto this, a certain machine named Mi can

only handle alimited number of time jobs. As soon as constraints are introduced to limit unmatched jobs and machines, the relationship between the group of tasks accomplished on the machine Mi and the group of machines handling Jj can be seen right away, providing a value of 0–1 for each job and machine.

4.3 Hybrid Tabu search

P|STsd|Cmax is ai complex combinatorial problem ii (Baker, 2). In addition, i

High computational advantage is one advantage of themeta-heuristic algorithm. The MIP model can generatea useful result in around five minutes with software with limited data. a three-phase hybrid TS method is suggested to overcome the issue.

Phase-1: a method to optimize the amount of completed jobs comes from Phase 1

Phase-2: The TS approach is used to produce better outcomes when shifting create span duties amongst various machines. The employment of the local search heuristic method results in a better arrangement of tasks for the machine-assisted removal and insertion.

Phase-3: The jobs of the machines are reordered, anda heuristic technique is used to decrease the maximumfinishing periods for the non-bottleneck machines.

4.4 Insertion and deleting procedures

The local neighbourhood notion is used via the basic heuristic algorithm (França et al., 4). Using the 2nd heuristic method, some solutions are capable of deviating from the local optimum.

When evaluating the immediate vicinity of a job, take into account utilizing q1 as an input parameter. Whencompared to a machine, a Jj's local neighbours are thetasks that are this machine's nearest predecessors and successors. The Jk with sjk less than q is the Jij's c successor, and the Ji with sij less than q is the Ji. I canonly choose the job with the fewest nearby neighboursas the best candidate for deletion. Each machine's localneighbourhood is identified, i.e., every task has m nearby properties. The sample below serves as a representation of the ideas. Think about the [sjk] matrix in Table 2. The matrix should include a new column where the element s0j stands in for the preparation time of Jj, the initial task in the order. Where $s_{j0}= 0$, i = 1,2,3,…,n.

Table 3: The matrix of s_{jk}

s_{ij}	0	1	2	3	4	5	6
0	–	30	32	54	8	38	42
1	0	–	35	25	47	7	88
2	0	21	–	76	38	86	25
3	0	8	38	–	30	32	54
4	0	48	48	23	–	8	45
5	0	23	28	18	36	–	61
6	0	74	82	61	25	38	–

Presume that 2 machines and the sequences allocated are:

M_1: – o– 2 – 5

M_2: – o – – 3 – – 1 – – 4 –- 6

local neighborhood method q = 23:

M_1: successors = {0, 5}; predecessors = {2, 5}

M_2: successors = {0}; predecessors = {3}

Selecting a job from one machine, then inserting it into another, is the fundamental concept underpinning each operation (insertion and deletion). The maximum finish time can be gradually lowered by accounting for the movement more than once.

Heuristic 1:

- Delete: Choose the work on the machine with the fewest nearby neighbors.
- Insertion: Calculate the number of nearby neighbors for the selected job on each machine that is capable. Choose the machine that has the most nearby neighbors after that.

Heuristic 2:

- Delete: Pick a job at random to be the delete job on the machine.
- Insertion: Put the selected task into a computer that can handle it at random. Once the deletion and insertion processes have been employed, heuristic 1 is applied to minimize the preparation time as much as conceivable. Furthermore, if the tasks of machine Mi are fixed, the entire setup time will have a significant impact on the maximum finishing time of these jobs because the operating times for these tasks on machine Mi are determined. The jobs setup times includes the Jj are shorter to q are the immediate neighbors for the. Accordingly, in theory, an order with shorter overall setup time is generated because each work on the machine Mi has more local neighbors. The elimination of the machine job that had the fewest close neighbours suggests that the remaining occupations require more close neighbours than the ones that were deleted. The insertion causes the deleted job to be moved to the machine with the maximum no. of adjacent neighbors. Or, to put it another way, these 2 models are employed to minimize the overall preparation time for jobs on the deletion and insertion-linked machines that have a lot of close neighbors.
- In comparison to randomly generated solutions, Heuristic 1 typically produces solutions that have a higher likelihood of reaching the feasible answer. Heuristic 2 can, however, prevent being stuck in the local optima. These two algorithms together will result in a search that is effective andof high caliber.

Glover (5, 6) created the TS global optimization meta-heuristic algorithm in 1989 and 1990. One significantdistinction from the 'hill' model.

In contrast to conventional climbing algorithms, the TS method may escape the local minima and offer superior solutions. The search procedure is equipped with a system that permits the objective to degrade. Additionally, the method allows the answer to leave the local optima by doing this. The measure with the lowest cost can determine the following generation. Itis necessary to accept an unimproved disturbance if a local minimum is the answer. Considering that the search selects the best movement of one iteration, the solution may potentially re-enter the local optimum from which it initially fled. The most recent iteration is forbidden (Tabu) and added to a list of Tabu in order to avoid this. The TS method has successfully addressed a number of combinatorial problems duringthe past few years by (Glover, 1990).

França et al.'s (1996) modification of the TS method for P|STsd|Cmax issues lacks the restrictions of job dispatch time and the elapsed time of allowing a work to be finished on a certain machine. The TS algorithms should be defined in termsof the characteristics of a single particular situation because they are open-ended. The following are the main characteristics that can improve how the TSalgorithm is implemented

a. The neighborhood arrangement.
b. first result.
c. The tabu tenure,
d. The stopping criterion.

In our implementation of the TS method to solve the R|rj, STsd|Cmax problem, a neighbor solution is obtained by shifting work of the occupied machine to another machine.

On the other hand, if it is too large, more neighbors may be prohibited and less neighbors may be offered,and the TS strategy might not work well since the restrictions are too tight to look for a better result. At the end of the segment, we'll talk about the best optionfor the Tabu tenure and other factors. The stop condition for the TS method is determined to be the number of iterations, T.

Phase 1 in the MIP model is the same as phase-1. Thesolution will be saved and used as the initial solutionin phase-2. Every machine's schedule, completiontime, and make span are all included in the solution. Assuming that machine is the busiest one since Phase-2 preparation times are not dependent of the machine,take into account the notations listed below (França et al., 4): Ci(Mi, Jj) is equal to the machine's completion time. If Jj is added to it, Mi. If Jj is taken out of the equation, Ci(MI, Jj) = the machine's completion time. Phase 2's detailed process is:

Let the counter to t = 0 in step 1.

Let the result counter to 1 in step two.

Step-3: As suggested in section 5.1, create the candidate solutions using Heuristics 1 and 2.

a) Using Heuristics 1 or 2, choose the Jj to be transported and gathered by the machine. Heuristic 1 will be used if s = 1; else, Heuristic 2 will be used.
b) Calculate the single machine's shortest completion time for each of the p randomly generated sequencesafter inserting the defined-a Jj:

$C^*(M_i^*, J_j) = \min\{C(M_i^*, J_j)|p$ order of the machine m machine $M_i^*\}$

c) After Jj has been deleted, determine which order on the occupied machine with p randomly generated orders has the shortest completion time under the premise that p is the total number of optimisation iterations.

$C^*(\bar{M}_i, J_j) = \min\{C(\bar{M}_i, J_j)|p$ order machine $\bar{M}_i\}$

d) Determine the maximum finishing time if Jj is transferred from to, as a: Ts = max|iM|C(Mi)
e) Return to action an at step 3 & build a new result ifthe replied solution is tabu.
Sets s = s + 1 in
f) Return to action at step 3 and go on to the next resultif s is 100. If not, proceed to step-4.

Step-4: Add the solution* to the Tabu list by selectingit as solution* = argminTs.

Step-5: Choose the machine that is now in use, then improve the best, the solution with the smallest make span, and the current solution.

Let t Equal t + 1 in step six. Go to step 2 if T, the required no. iterations, has not been arrived. If not, finish the optimization and export the result.

A single-machine scheduling issue could be used to describe the issue in phase 3. Each non-bottleneck machine's maximum completion time should be as short as possible.

Consider the following actions for every non- bottleneck machine:

Step-5.1: If the no. of completed jobs is not lesser zero, carry out move 2; disregard this machine and take the next into consideration.

Create p random result (p task order) and choose the quickest finish time in move 2-2.

The parameter values are accompanied with comments. The formula p = min 100, ni!, where ni is the number of tasks on the machine that is the subject of the optimisation, determines the number of arbitrarily produced sequences, p, needed for a single machine to provide a least finishing time. The Tabu tenure can be selected based on particular instances, and in this case, 12 is chosen as the Tabu tenure. T is predetermined to be 1.2n iterations. The proposed method has a respectable numerical efficiency after being evaluated on 30 computers and 90 jobs.

These qualities are listed below:

- AppId: the unique job identifier provided by anapplication.
- Pppid is a job attribute that will be utilized tocalculate setup time.
- Process time: How long it takes a machine to do a job
- The gas type used to process a job.
- dispatch time: the anticipated time of release
- The machine identification value is EqpId.
- Running AppId: This is the value used to identify a job that is currently running on the machine. This feature is used to calculate the preparation time for the first work on the machine.
- Gs Running. This characteristic is used to calculate the setup time for the first task on the machine.
- Machine accessibility: the duration of a machine'savailability
- RdReason: the justification for the impossibility of this machine-job combination. In the experiment, all other rows will be erased except for those where RdReason is empty.
- Expired time: The amount of time that has passedsince this machine and job were combined; if thejob cannot be completed by this time, it willexpire.
- The job already running on the system and its successor determine the setup time:
- If the pppid for these two jobs is same, the preparation time is 0 seconds.
- If the three jobs have various time intervals but the same gap, preparation time is 65 seconds.
- Preparation time is 911 s if t 2 jobs go well.

5. Computational Results

The MIP model is used to address the issue with the first two objectives, and the hybrid TS method

is used to resolve the issue with all three objectives.

Table 4 MIP and TS solution results

CI	NJ	T_{MIP}	T_{TS}	Z_{MIP}	Z_{TS}
1	73	304.73	19.99	13,103	12,300
2	77	307.3	23.37	1 1,444	1 1,444
3	76	304.34	24.94	20,064	13,232
3	77	304.33	18.83	13,842	9702
4	74	303.87	19.42	13,687	8794
6	72	302.24	12.39	13,360	9034
7	69	303.73	9.33	13,244	11,290
8	68	303.37	9.83	14,681	13,964
9	89	306.33	20.09	14,173	12,632
10	80	310.01	44.23	20,136	9320
Synthesized data	79	304.34	23.38	14,793	11,293

The greatest number of jobs that may be finished, the typical computation time for different scenarios utilising the MIP and TS techniques, and the objective value of the MIP and TS after phase 2 are all shown in Table 3. The average computing time when these two techniques are used is the sum of phases 1 and 2. In Fig. 2, the model's maximum completion times are contrasted. The equation is used to compute the relative improvement:

$$(Z_{MIP} - Z_{TS}) / Z_{MIP} \quad (49)$$

The terminology used:

Cli	index
NJi	Max. no. of processed jobs
Ti_{MIP}	Average computational time through MIP
Ti_{TS}	Average computational time through TS algorithm (s)
Zi_{MIP}	MIP ans (s)
ZI_{TS}	TSans (s)

Table 5 The results using TS

MI	Z_2	Z_3	R_{phase3}
2	7777	6662	0.127
3	1076	1076	0
7	7317	7317	0
12	1 1,030	1 1,030	0
13	10,676	10,676	0
17	10,326	10,326	0

16	3673	7303	0.097
16	6766	7300	0.263
17	9770	9770	0
20	3666	3666	0
22	6172	6026	0.132
23	3621	3621	0
29	6370	7970	0.076
32	6362	6136	0.072

Table 5, the relative improvement is computed.

$$R_{phase3} = \frac{Z_2 - Z_3}{Z_2} \quad (50)$$

The terminology used in Table 5 is as follows:

MI	Machine order
Z_2	Maximum finishing time of the machine in phase-2
Z_3	Maximum finishing time of the machine in phase-3
R_{phase3}	Relative enhancement starting phase-2 to phase-3

6. Conclusions

The sequence-dependent setup times eij, STsd, and Cmax should be used to model the ion implantation of scheduling as a UPMS problem. These setup times are influenced by the job release timings and the lapsed times of a task's permission when carried out on a particular machine. The other a hybrid TS method that satisfies all three ofthe problem's objectives. The MIP technique l's effectiveness and efficiency are demonstrated to be lower. Since the process times cannot be ascertained accurately, process times can be considered stochastic variables for dynamic job shop scheduling and unrelatedmachine scheduling considering preventivemaintenance as a future scope of this study.

References

1. Chang H.C. et al. “Solving the Flexible Job Shop Scheduling Problem with Makespan Optimization by Using a Hybrid Taguchi-Genetic Algorithm”. IEEE Access 3(2015):1740-1754.
2. PaprockaI. “Evaluation of the Effects of a Machine Failure on the Robustness of a Job Shop System- Proactive Approaches”. Sustainability 11.1(2019):65.
3. MH Abd Elrahman Elgendy., et al. “Optimizing Dynamic Flexible Job Shop Scheduling Problem Based on Genetic Algorithm”. International Journal of Current Engineering and Technology 7 (2017):368-373
4. Ben AliK., etal. Stochastic cases of the dynamic job shop problem based on the genetic algorithm to minimize theCmax. CoDIT. IEEE, Barcelona, Spain (2017): 760-765.
5. Graves G., et al. “Scheduling maintenance and semi- resumable job son single machine”. Naval Research Logistics 46 (1999): 845-862.

6. Lee W-C., et al. "Multi-machine scheduling with deteriorating job and scheduled maintenance". Applied Mathematical Modelling 32.3(2008):362-373.
7. Brandolese M., et al. "Production and maintenance integrated planning". International Journal of ProductionResearch 34.7 (1996):2059-2075
8. Allahverdi, A., Ng, C., Cheng, T. E., & Kovalyov, M.Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3),985-1032.
9. Baker, K.R. (1974). Introduction to sequencing and scheduling. John Wiley & Sons.
10. Bradley, S.P., Hax, A.C., & Magnanti, T.L. (1977). Applied mathematical programming. Addison-Wesley.
11. França, P.M., Gendreau, M., Laporte,G., & Müller, F.M. (1996). Atabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. International Journal of Production Economics, 43(2–3), 79–89.
12. Glover, F. (1989). Tabu search-part I. ORSA Journal onComputing, 1(3),190–206.
13. Glover, F.(1990). Tabu search-part II. ORSA Journal onComputing, 2(1), 4–32.
14. Ben AliK., et al. "An Improved Genetic Algorithm withLocal Search for Solving the DJSSP with New DynamicEvents". ETFA. IEEE, Turin, Italy(2018):1137-1144.
15. Yang Z Dragan D · Jun N., "Maintenance scheduling inmanufacturing systems based on predicted machine degradation". Journal of Intelligent Manufacturing 19.1(2008):87-98.
16. Zhang, L., Deng, Q., Lin, R., Gong, G., & Han, W., A combinational evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect. Expert Systems with Applications, 175, Article 114843(2021).
17. Teoh YK., et al. "IoT and Fog Computing based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 using Machine Learning". IEEE Internet of Things Journal (2021).
18. Yang Z., et al. "Maintenance scheduling in manufacturing systems based on predicted machine degradation". Journal of Intelligent Manufacturing 19.1(2008):87-98.
19. Zhang, L., Deng, Q., Lin, R., Gong, G., & Han, W., A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect. Expert Systems with Applications, 175, Article 114843 (2021).