IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 2, Ser. 1 (Mar. – Apr. 2024), PP 52-58
www.iosrjournals.org

A Heuristic For M/M/1 Scheduling Problem With Common Due Date

Ilaya Ufouma[1] and Tsetimi Jonathan[2]
[1,2]Department of Mathematics, Delta State University, Abraka, Delta State, Nigeria
E-mail Addresses: [1]ufoumailaya@gmail.com, [2]jtsetimi@delsu.edu.ng

Abstract
In the manufacturing industry, the scheduling of tasks on a single machine with a common due date is a common problem faced by many manufacturers. The objective is to minimize the tardiness of the jobs that need to be completed by the due date. The study was aimed at exploring the single machine scheduling problem with a common due date and propose efficient solutions to solve it. The study reviewed relevant literature, analyze the problem characteristics, formulate a mathematical model, propose efficient algorithms, evaluate their performance, and compare them with existing methods to demonstrate their effectiveness. It is observed that among the schedules used to compare the sum of total earliness and tardiness costs in this study with optimal schedule of { J_4, J_6, J_1, J_2, J_5, J_3} gave a penalty of 90 days. Although the difference 10 days is not much but it can help to increase inventory level, causes losses owing to deterioration if the wrong optimal schedule is choose and in general indicate sub - optimal resource allocation and utilization. This study has been able to establish that the simple heuristic algorithm outlining the procedure for minimizing the sum total earliness and tardiness costs in an n-job single machine scheduling problem. We also established that scheduling problems arise from real-life situations and as such are very complex, and as such are very complex, and can be optimized with the most efficient algorithm manually or computationally. This study has also been able to establish that the order in which jobs are processed on each machine is the same and the number of feasible schedules is n!.

Date of Submission: 04-04-2024 Date of Acceptance: 14-04-2024

Keywords: Scheduling Problem, Heurisic Method, n-job single machine scheduling problem, Optimal value.

I. Introduction

In today's complex industrial world, many business problems that need to be solved or optimized are scheduling problems. A manufacturing firm producing multiple products, each requiring many different processes and machine facilities for completion must find a way to successfully manage resources in the most efficient way possible. The decision maker is faced with the problem of designing a production or job schedule that promotes on time delivery and minimizes objectives such as the flow time and completion time of a product. Due to the complexity of optimization as a result of objective constraints, some production firms consider all their customers to be of equal importance.

For instance, suppose that there are a number of jobs to be processed and that one is for a customer of exceptional importance whose goodwill must be maintained at all costs. In this case, the firm is extremely likely to decide that his job should be rushed through and finished first, no matter the consequences or effect it will have on other jobs. Thus, a scheduling problem is complicated by precedence constraints. In general, this limit the choice of schedule by demanding that certain subsets of jobs is processed in a given order. There are many other practical scheduling problems which precedence constraints arise, but it will be sufficient to illustrate just two more here; first we consider a situation where any adjustment to a machine settings may require a substantial time to take effect; perhaps certain components need to warm up or cool down. In such a case, it makes sense to process together any group of jobs that require similar settings, because the change over-time between the jobs will be slight.

Secondly, we consider the situation of a scheduling problem that involves a computer program. Suppose that one program produces an output file that a second requires an output file that a second requires as input data. Then, obviously the schedule must ensure; that the first is fully executed before the second is begun. Aside from scheduling problems with precedence continues, a frequently occurring scheduling problem is one g^z processing a given number of jobs on a specific number of machines. This class g^z problem also referred to by many as dispatching or sequencing is categorized as NP-hard. The desire to process the jobs in a specific order achieve some objective function is what creates a problem that remains largely unsolved. The actual

situations that give rise to scheduling problems are wide and varied. This includes the following: Simple machine Scheduling problem, multiple machine scheduling problems and manpower scheduling problem.
A general n jobs m machines scheduling problem can be stated as follows:

Given n jobs to be processed on m machines in the same technological order, the processing time of job i or machine j being tij (i =1, 2, …….., n; j = 1, 2, …….., m), it is desired to find the order (schedule or sequence), which these n jobs should be processed on each of the m machines so as to optimize (minimize or maximize) a well-defined measure of some objectives (such as production cost, number g^z late jobs, etc.) This problem in general is given (n!) as possible schedule. Even for problems as small as n = m = s, the number of possible schedules is so large that a direct enumeration is economically impossible. For n = s, and m = z, we have 1, 728, 000 possible schedules.

However, for a simplified version where it is the same, that is, the number of jobs no matter how many is processed on a single machine making m = 1, we have the number of feasible schedules reduced to n!.Single machine scheduling problem bear complex computations and the analysis of such problems is important and for a better understanding of the problem. Among single machine problems those related to earliness and tardiness is more important. Completing jobs or task earlier than their due dates should be discouraged as much as completing jobs or tasks later than their due dates. In real world, since a customer expects to receive the product on a specific date, scheduling based on the due date is also an important task in the production planning. Earliness leads to inventory and maintenance carrying cost while tardiness leads to customer's dissatisfaction and losing goodwill and reputations.

II. Literature Review

Job scheduling or sequencing has a wide variety of applications, from designing the product flow and order in a manufacturing facility to modeling queues in service industries (Vargas and Calvo, 2018). It is indeed a useful subject that is still being actively researched. Most researchers focused on special cases in which situational restrictions are imposed. Tsetimi (2013) showed that hybrid Heuristic approach to single machine scheduling problems using learning effect and precedence constraints. Cheng *et al.* (2004) studied a single machine due date assignment scheduling.

In scheduling problem, as the numbers of jobs and machines get larger, the number of possible schedules also increases. According to Nahmias (1997), for the n-jobs and m-machine problem, there are $(n!)^m$ possible schedule. For more than two machine scheduling problem, the problem becomes a NP-hard problem. (Gapta and Chauhan, 2015). According to Tsetimi (2010), the prefix NP which is a non-polynomial implies that the computational time involved is exponential rather than a polynomial function of the variables involved. He also went ahead to say that if the rate of change is non-polynomial as the variable involved becomes large, the problem is said to be NP. The problem is said to be NP-hand when all sequence available for the scheduling are so many that it would be impractical to enumerate the problem, and this has been a basis for development of heuristics.

A heuristics approach is a procedure in which a set of rule is systematically applied. (Adam and Ebert, 2003). According to Stevenson (2007), a heuristics is an intuitive approach that yields an optimal solution. According to Tsetimi (2010), heuristics often involves the application of elementary rules in order to achieve an optimal or sub optimal or a near optimal solution to the problem. The methods that seek to achieve a feasible solution close to the optimal in a reasonable computation time can be classified into heuristics and metaheuristics (Yenisey and Yagmahan, 2014). Pereira (2011) opined that the solution for problem using optimum method usually demand high computational time, making the search for optimal solution not viable.

Tsetimi (2010), emphasizes the need for Heuristics, he pointed out one way of avoiding complete enumeration in scheduling is to developing a recursive solution by dynamic programming. In complete enumeration, the numbers of computation in these recursive solutions soon become too many and time consuming. Sometimes, heuristics give solution that do not necessarily have to be close to optimal but they provide flexibility for the scheduler and give initial solution in a reasonable time.

Generally in scheduling problems it is assumed that machines are continuously available over the planning horizon. However, this assumption may not be true in many practical situations (Azadeh*et al.,* 2013). For instance, a machine may not be available during the planning horizon due to maintenance activities, tool changes, or breakdowns. Since, machines require preventive and curative maintenance, operators take breaks, and worn out parts require changing. Not only managers are increasingly faced with the costs caused by the temporary unavailability of resources, but also they are constantly concerned with difficult decisions regarding balancing resources' unavailability and production (Yazdani*et al.,* 2016).

III. Established Methods Of Solving Single Machine Scheduling Models Shortest Processing Time (Spt)

The shortest processing time can be used to minimize the mean flow time in a single machine scheduling problem. Hence, the SPT rule can be used to solve any scheduling problem as long as the number of jobs or task is known and the processing time and due dates for each jobs is known.

Thus, in Tsetimi (2010), the SPT rule can be used to solve the $n/1/\overrightarrow{C}$, $n/1/\overrightarrow{W}$, $n/1/\overrightarrow{L}$, and $n/1/\overrightarrow{N}$, classes of processing times. In the SPT rule if P_k^i is the processing time of the job that is processed *kth* in the processing sequence, then we can sequence the jobs such that,

$$P_1^i \leq P_2^i \leq P_3^i \leq P_4^i \leq \cdots \leq P_n^i$$

Earliest Due Date

The Earliest Due Date (EDD) Scheduling rule can be used to solve the $n/1/\ L_{max}$ and $n/1/\ T_{max}$ classes of scheduling problems. To use the EDD rule, we can make use of the following steps.

1. Define Job Data: For each job, identify it's processing time and due date
2. Sort Jobs: Arrange the jobs in ascending order based on their due date. Jobs with the earliest due dates come first.
3. Schedule Jobs: Execute or schedule the jobs in the order determined by the sorted list in step 2, start with the job having the earliest due date.
4. Completion Time Calculation: Calculate the completion time for each job. The completion time is the sum of the processing times of all previously scheduled jobs and the processing times of the current job.

5 Evaluate Performance: Assess the performance of the schedule based on criteria such as total completion times or total tardiness.

6. Repeat the Process: Repeat the process until all jobs are scheduled.

The primary focus of EDD is to prioritize jobs with the earliest due date ensuring that jobs are completed closer for their respective deadline.

This method is intuitive and easy to understand but may not always result in an optimal solution, especially when considering other performance metrics.

Based on the steps above, supposed d_k^i is the due dates for the job that is processed *kth* in the processing sequence, we arrange the jobs such that;

$$d_1^i \leq d_2^i \leq d_3^i \leq d_4^i \leq \cdots \leq d_n^i$$

Lawler's Algorithm

This is an algorithm that was developed by Lawler (1973) to deal with rather more general precedence constraints. In this algorithm, we shall simply be constrained to process certain jobs before, but not necessarily immediately before certain other jobs. Lawler's algorithm minimize the maximum cost of processing job, where this cost has a general form denoted by $y_i\ C_i$ for each job $J_i$and is taken to be non-decreasing in the completion time C_i thus the algorithm minimizes

$$\max_{1 \leq i \leq n} (y_i\ C_i)$$

Because of the nature of the non-decreasing form of $y_i\ C_i$, it follows immediately that this performance measure is regular. To develop the Lawler's algorithm, we need the following theorem.

Theorem 1

Consider the $n/1/\ \max_{1 \leq i \leq n} (y_i\ C_i)$ scheduling problem with precedence constraints. Let V denote the subset of jobs which may be performed last, that is, those jobs which are not required to precede any other. Note that the final job in the schedule must complete at $\tau = \sum_{i=1}^{n} P_i$, Let J_k be a job in V, that is of all the jobs that may be performed last J_k incurs the least cost. Then, there is an optimal schedule in which J_k is scheduled last.

Proof:

Suppose S is an optimal schedule with J_k not last. We assume that S is compatible with the precedence constraints. Then S has the form

$$(A,\ J_k, B, J_l)$$

Where J_l is the last job under S and A and B are subsequences of the other (n – 2) jobs. It is should be noted that A and B could be empty. Consider the new sequence

$$S^{/} = (A, B\ , J_l, J_k)$$

Since S obeyed the precedence constraints and J_k may be last, $S^{/}$ is feasible. Remember that $(y_i\ C_i)$ is non-decreasing in C_i . Since all completion times of jobs other than J_k have decreased in passing from S to $S^{/}$ no cost other than for J_k can have increased. By construction J_k is chosen such that

$$y_k(\tau) = \min_{J_i in\ V} \{y_i(\tau)\}$$

$$\leq y_i(\tau)$$

Smith Algorithm

The smith algorithm is an algorithm that finds schedule that are efficient and although not directed to the immediate solution of the problem is nevertheless instrumental in the construction of efficient schedules.

The following are the necessary steps for the Smith's Algorithm
Step 1: Set $k = n, \tau = \sum_{i=1}^{n} p_i$; $U = \{J_1, J_2, \ldots, J_n\}$.
Step 2: Find $J_{i(k)}$ in U such that $d_{i(k)} \geq \tau$ and $p_{i(k)} \geq p_i \; for \; all \; J_i \; in \; U \;\; such \; that \; d_i \geq \tau$
Step 3: Decrease k by 1; decrease $\tau \;\; by \; p_{i(k)}$; delete $J_{i(k)}$ from U
Step 4: If there are some jobs to schedule, that is, If k≥ 1, go to Step 2,. Otherwise, stop with the optimal processing sequence, $\{J_{i(1)}, J_{i(2)}, \ldots, J_{i(n)}\}$.

IV. The General Single Machine Problem With Common Due Date

This problem can be mathematically formulated as

$$minimize \; F = \sum_{i=1}^{n} \{\alpha_i \max(d - C_i, 0) + \beta_i \max(C_i - d, 0)\}$$

$Where \; C_i, i = 1, 2, 3, \cdots, n$ is the completion time for Job I, d is the common due date and $\alpha_i$and β_i are the unit penalty costs associated with earliness and tardiness respectively.

The objective here is to use the general single machine problem with common due date mathematically formulated formula to find which of these 6 schedules the optimal schedule as well as the optimal value. The last schedule with the minimum optimal value becomes the optimal schedule. The optimal value is given by the following for each schedule *S*.

$$\bar{F} = S_i = \sum_{i=1}^{n} |p_i - d|$$

V. Numerical Perspective Of The Heuristic Algorithm

Heuristic Algorithm Method

Due to the nature of the cumbersome and tedious intentions involved when dealing with large number of jobs which invariably results to millions or billions of schedules, a convenient Algorithm is formulated in this chapter called the Heuristic Algorithm.

Given n-jobs to be processed on a single machine, the processing time of job i being ti; for i = 1, 2…, n. It is assumed that all jobs are ready for processing at time zero and have the same common due date (deadline), D. The problem is to find the order (schedule) in which these n-jobs should be processed so as to minimize the sum of total earliness and tardiness costs. The common due date, D is assumed to be less than the total processing time.

Initial step O: Sort and number the n-jobs in non-increasing order of processing time ti, (i =1, 2, .., n) such that $t_1 \geq \ldots \geq t_{1-1} \geq t_i \geq t_{i+1} \geq \ldots \geq t_n$
In general, we can represent the jobs in tabular form.

J_i	J_1	J_2	… J_{i-1}	J_i	J_{i+1}	…	J_n
t_i	t_1	t_2	… t_{i-1}	t_i	t_{i+1}	…	t_n

We evaluate;

- Total processing time, $T = \sum_{i=1}^{n} t_i$
- $T_0 = T - D_1$ and $E_0 = D$
- We also introduce two empty sets as the initial sets of schedules, S_0^E and S_0^T called initial schedule set based on due date S_0^E and initial schedule set based on processing time of job.

Step 1: First iteration, for i = 1
Consider the first job J_1 with processing time $t_1 = \max(t_i, i = 1, 2, \ldots, n)$
Then, set $T_1 = T_0$, and $E_1 = E_0$
Now;

- If $T_1 < E_1$, Set $S_1^E = S_0^E + \{J_1\}$ and $S_1^T = S_0^T$
- If $T_1 \geq E_1$, set $S_1^E = S_0^E$ and $S_1^T = S_0^T + \{J_1\}$

Step i: ith iteration, for i = n
We consider the last job J_n, with processing time t_i, such that ($1 < i \leq n$)
Also, we make the following consideration;
- If previous job, $J_{i-1} \in S_{i-1}^E$, then we evaluate $T_i = T_{i-1}$ and $E_i = E_{i-1} - t_{i-1}$
- If previous job $J_{i-1} \in S_{i-1}^T$, then we evaluate $T_i = T_{i-1} - t_{i-1}$, and $E_i = E_{i-1}$

Thereafter, we assigned the following decision to the schedule set.
- If $T_i < E_i$, Set $S_i^E = S_{i-1}^E + \{J_i\}$ and $S_i^T = S_{i-1}^T$
- If $T_i \geq E_i$, set $S_i^E = S_{i-1}^E$ and $S_i^T = S_{i-1}^T + \{J_i\}$

The iteration terminates when all jobs have been assigned to either S_n^E or S_n^T.
Now, for the scheduling decision; we check so that;
S_n^E = {jobs in non-increasing order processing time}
S_n^T = {jobs in non-decreasing order processing time}
The optimal schedule, $S^* = \{ S_n^E \text{ and } S_n^T \}$
In other words, jobs are scheduled to their sequence in S_n^E and followed by S_n^T.

We also note here that this algorithm or procedure only involve n enumerations or iterations as compound to n! Possible schedules.

Numerical Illustration Of The Hueristic Algorithm Method

Consider the 6/1/m problem with data below and the constraints that J_1 must precede J_2 which must in turn precede both J_5 and J_6 such that we have the following scheduling table with common due date of 20

When we sum up the total number of processing time, If the total sum of processing time is less than the common due date.

Hence we create a new processing time by sum the old processing time and the respective due dates. Thus we have the tabular form as follows

Now Total Processing Time T
$T^* = \sum t * = 58$
And the
$T_o = T^* - D = 38$
$E_o = 20$
We introduce two empty sets
$S_o^E = \emptyset$ and $S_o^T = \emptyset$

First iteration

Total Processing time T
$T = \sum_{i=1}^{6} ti = 2 + 3 + 4 + 3 + 2 + 1 = 15$
Now $T_1 = T - D$, :. $58 - 20 = 38$
$E_o = D = 20$
When First iteration for i = 1
The new processing time = 5
$T_1 = T_o = 38$, $E_1 = E_o = 20$
Solve
$T_1 > E_1$, $S_1^T = S_o^T + [J_1] = \{J_1\}$
And $S_1^E = S_o^T = [\ ]$

Second Iteration i = 2

We consider Job J_2 with new processing time $t_2 = 9$
And since $J_1 \epsilon S^T$ we compute
$T_2 = T_1 - t_1 = 38 - 5 = 33$ and
$E_2 = E_1 = 20$
$T_2 > E_2$,:. $S_2^t = S_1^t + [J_2] = \{J_1, J_2\}$
And $S_2^E = S_1^t = [\ ]$

Third Iteration i = 3

We consider Job J_3 with new processing time,
$t_3 = 13$ and solve
$J_2 \epsilon S_2^t$ we compute
$T_3 = T_2 - t_2 = 33 - 9 = 24$
$E_3 = E_2 = 20$
Since $T_3 > E_3$,:. $S_3^t = S_3^T + [J_3]$
$= \{J_1\ J_2\ J_3\}$

And $S_3^E = S_2^E = [\ ]$

Fourth Iteration i = 4

We consider Job J_4 with new processing time J + 4 = 10 and since $J_3 \epsilon S_3^t$

We have

$T_4 = T_3 - t_3 = 24 - 13 = 11$

And $E_4 = E_3 = 20$

Now $T_4 < E_{3,:} S_4^t = S_3^t = [J_1, J_2, J_3]$

And $S_4^1 = S_3^1 + [J_4] = [J_4]$

Fifth Iteration i =5

We consider Jobs J_5 with new processing time $t_5 = 13$ and since

$J_4 \epsilon S_4^E$ we compute

$T_5 = T_4 = 11$ and

$E_5 = E_4 - t_4 = 20 - 10 = 10$

Now $T_5 > E_5$

$S_5^t = S_4^t + [J_5] = [J_1\ J_2\ J_3\ J_5]$

And

$S_5^t = S_4^E = [J_4]$

Sixth Iteration i = 6

We consider Job J_6 with new processing time $t_6 = 8$ and since

$J_5 \epsilon S_5^t$ we compute

$T_6 = T_5 - t_5 = 11 - 13 = -2$ and $E_6 = E_5 = 10$

Now $T_6 < E_6$

$S^T = S^T = \{J_1\ J_2\ J_3\ J_5\}$ and

$S^T_6 = S^E_5 + \{J_6\} = \{J_4\ J_6\}$

This is the end of iteration since all Jobs have been assigned to other S_6^T or S_6^T

Now, we observe that jobs in $S_6^E = \{J_4, J_6\} = \{10, 8\}$ are in non-increasing order as required, but jobs in $S_6^T = \{J_1, J_2, J_3, J_5\} = \{5, 9, 13, 10\}$ are not in the required non-decreasingly order. Thus, we reunite $S_6^T = \{J_1, J_2, J_5, J_3\}$, which is now the required non-decreasing order.

VI. Conclusion

The Heuristics Algorithm quickly provides a schedule, which sometimes may not yield the optimal solution or be the most cost efficient when it comes to larger number of Jobs or task. Heuristic offers practical and efficient solutions for scheduling problems while they may not guarantee optimality, their speed and adaptability makes them valuable tools for managing 'less 'complex real world situation. Based on the numerical results from the study, it is observed that among the schedules used to compare the sum of total earliness and tardiness costs is the optimal schedule of the Hybrid Algorithm by Tsetimi (2009) which is *{J_1, J_2, J_4, J_6, J_3, J_5}* that was obtained by the learning effect methods on the processing times; which gives a penalty of 100 days compared to the Heuristic Algorithm presented in this paper with optimal schedule of

{J_4, J_6, J_1, J_2, J_5, J_3} which gives a penalty of 90 days. Although the difference 10 days is not much but it can help to increase inventory level, causes losses owing to deterioration if the wrong optimal schedule is choose and in general indicate sub - optimal resource allocation and utilization.

References

[1] Alidaee, B., N.K. Womer, Scheduling With Time Dependent Processing Processing Times: Review And Extensions, Journal Of The Operational Research Society 50 (1999) 711–720.

[2] Azadeh A, Sheikhalishahi M, Firoozi M, Khalili S. An Integrated Multi-Criteria Taguchi Computer Simulation-Dea Approach For Optimum Main-Tenance Policy And Planning By Incorporating Learning Effects.Int J Prodres2013;51:5374–85

[3] Cai-Min Wei A,⇑, Ji-Bo Wang B, Ping Ji (2012). Single-Machine Scheduling With Time-And-Resource-Dependent Processing Times. Applied Mathematical Modelling 36 (2012) 792–798.

[4] Chen, B., & Lin, B. (2015). A Variable Neighborhood Search Algorithm For The Single-Machine Scheduling Problem With A Common Due Date. European Journal Of Operational Research, 246(2), 587-596.

[5] Hua Gong ,1,2yuyan Zhang,1 And Puyu Yuan (2020). Scheduling On A Single Machine And Parallel Machines With Batch Deliveries And Potential Disruption. Complexity20:1-10.

[6] Huang, X., M.-Z. Wang, Parallel Identical Machines Scheduling With Deteriorating Jobs And Total Absolute Differences Penalties, Applied Mathematical Modelling 35 (2011) 1349–1353.

[7] Janiak, A., M.Y. Kovalyov, Scheduling In A Contaminated Area: A Model And Polynomial Algorithms, European Journal Of Operational Research 173 (2006) 125–132.

[8] Kacem, I., Hammadi, S., &Loukil, T. (2013). A Branch-And-Bound Algorithm For The Single Machine Scheduling Problem With Common Due Date. Journal Of Scheduling, 16(1), 73-83.

[9] Laalaoui Y, M'hallah R. A Binary Multiple Knapsack Model ForSinglemachine Scheduling With Machine Unavailability.Computoper Res2016;72:71–82.

[10] Lee, W.C., C.-C. Wu, C.-C. Wen, Y.-H. Chung, A Two-Machine Flowshopmakespan Scheduling Problem With Deteriorating Jobs, Computers & Industrial Engineering 54 (2008) 737–749.

[11] Lee, W.C., C.-C. Wu, Y.-H. Chung, H.-C. Liu, Minimizing The Total Completion Time In Permutation Flow Shop With Machine-Dependent Job Deterioration Rates, Computers & Industrial Engineering 36 (2009) 2111–2121.

[12] Li, Y., G. Li, L. Sun, Z. Xu, Single Machine Scheduling Of Deteriorating Jobs To Minimize Total Absolute Differences In Completion Times, International Journal Of Production Economics 118 (2009) 424–429.

[13] Liao, C. J And Chen W.J. (2003). Single Machine Scheduling With Periodic Maintenance And Non Resumable Jobs. Computers And Operations Research, 30(9):1335-1347.

[14] Lin, Sw., Ying, Kc. Single Machine Scheduling Problems With Sequence-Dependent

[15] Ranconi, D.P. And Kavamura, M.S. (2010). The Single Machine Earliness And Tardiness Scheduling Problem; Lower Bounds And Branch Algorithm Computational And Applied Mathematics, 29(2): 107-124.

[16] Rapine C, Brauner N, Finke G, Lebacque V. Single Machine Schedulingwith Small Operator-Non-Availability Periods.J Sched2012;15:127–39.

[17] Tang, L., H. Gong, J. Liu, And F. Li, "Bicriteria Scheduling On A Single Batching Machine With Job Transportation And Deterioration Considerations," Naval Research Logistics (Nrl), Vol. 61, No. 4, Pp. 269–285, 2014.

[18] Tang, L., P. Liu, Two-Machine Flowshop Scheduling Problems Involving A Batching Machine With Transportation Or Deterioration Consideration, Applied Mathematical Modelling 33 (2009) 1187–1199.

[19] Tsetimi,J.(2013) The Journal Of The Nigerian Institution Of Production Engineer/ Vol.14 March 2013. Pp.199-207

[20] Tsetimi. J. And Omosigho S. E. (2003). Single Machine Earliest Due Date Scheduling With Tardy Jobs. Journal Of The Nigerian Institution Of Production Engineers. Vol. 8, No.1 Pp 110 - 117.

[21] Tsetimi. J. And Omotor, D.G. (2011). A Permutation Schedule For Two-Machine Flow Shop Scheduling Problem With Due Dates Included. Journal Of Social And Management Sciences 6(1):1-5.

[22] Uzorh, A.C. And Innocent, N. (2014). Solving Machine Shops Scheduling Problem Using Priority Sequencing Rules Techniques. International Journal Of Engineering And Science, 3(6):15-22.

[23] Vargas J, Calvo R. Joint Optimization Of Process Flow And Scheduling In Service-Oriented Manufacturing Systems. Materials (Basel). 2018 Aug 29;11(9):1559.

[24] Wan, L. And A. Zhang, "Coordinated Scheduling On Parallel Machines With Batch Delivery," International Journal Of Production Economics, Vol. 150, No. 4, Pp. 199–203, 2014.

[25] Wang, D.Y., O. Grunder, And A. E. Moudni, "Integrated Scheduling Of Production And Distribution Operations: A Review," International Journal Of Industrial And Systems Engineering, Vol. 19, No. 1, Pp. 94–122, 2015.

[26] Wang, G. And T. C. E. Cheng, "Parallel Machine Scheduling With Batch Delivery Costs," International Journal Of Production Economics, Vol. 68, No. 2, Pp. 177–183, 2000.

[27] Wang, J.B. And Wang, M.T. (2012) Single Machine Scheduling With Non-Linear Deterioration. Opt. Letter, 6:87-98.

[28] Wang, J.B., C.T. Ng, T.C.E. Cheng, Single-Machine Scheduling With Deteriorating Jobs Under A Series-Parallel Graph Constraint, Computers And Operations Research 35 (2008) 2684–2693.

[29] Wang, J.B., J-J. Wang, P. Ji, Scheduling Jobs With Chain Precedence Constraints And Deteriorating Jobs, Journal Of The Operational Research Society (2010), Doi:10.1057/Jors.2010.120.

[30] Yang, S.H., J.-B. Wang, Minimizing Total Weighted Completion Time In A Two-Machine Flow Shop Scheduling Under Simple Linear Deterioration, Applied Mathematics And Computation 217 (2011) 4819–4826.

[31] Yang, S.J., Single-Machine Scheduling Problems With Both Start-Time Dependent Learning And Position Dependent Aging Effects Under Deteriorating Maintenance Consideration, Applied Mathematics And Computation 217 (2011) 3321–3329.

[32] Yazdani M, Khalili Sm, Jolai F. A Parallel Machine Scheduling Problemwith Two-Agent And Tool Change Activities: An Efficient Hybrid Metaheur-Isticalgorithm.Int J Computintegr Manuf2016:1–14.