Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

The Deviation Issue and Total Completion Time on a Single Machine Scale Subject to Decreasing and Growing Linear Deterioration are Theoretically Addressed

Mohammed A. Raheem[1] *, Hanan Ali Chachan[2].

[1,2]Mathematics Department, College of Science, Mustansiriyah University, Bagdad, IRAQ.

*Corresponding Author: Mohammed A. Raheem.

DOI: https://doi.org/10.31185/wjps.569
Received 10 October 2024; Accepted 15 November 2024; Available online 30 December 2024

ABSTRACT: We addressed in this research the schedule of activities that are subject to the condition of linear degradation in declining and growing rate states, respectively, with the goal of guaranteeing that the primary process's duration would be fixed (ρ) and that its implementation would take place in a single machine. The objective was to find the optimal schedule for calculating the total completion time $\sum_{i=1}^{\acute{n}} C_i$, and the optimal schedule for the deviation problem (sum of squares of the difference between the tasks' total required time and the previously established due date time. where the objective is to minimize the squared deviation of job completion times from a due date), two theories that state as the Λ-shaped scheduling theory with increasing rate λ_i of jobs and the v-shaped scheduling theory with decreasing rate $đ_i$ of jobs satisfy the conditions $0 < đ_i < 1$ and $đ_i < \frac{1}{\acute{n}-1}$, have been demonstrated in situations where the base is implemented at a constant basic processing time .

KEYWORDS: Single machine, linear deterioration, total completion time, Λ-shaped scheduling theory, v-shaped scheduling theory.

1. INTRODUCTION

Production scheduling is a crucial area where uncertainty plays a significant role, as is true for any other manufacturing-related discipline. The scheduling system's randomness may result from a variety of factors, including inconsistent processing speeds, malfunctioning machinery, and inaccurate client due date information. Some negative effects would include system instability, excess inventory, client unhappiness from missing the deadline, and-most importantly-loss of money if the schedule parameters unquestionably exist , see [1],[2], [3].

Considering the aforementioned, the classical scheduling, it is widely acknowledged that job processing times are functions that are independent of their start timings. We address this assumption in practice. However, we frequently come into environments where processing times gradually grow. Such a decline in processing time in a production scenario might happen, for example, if the machine starts to lose efficiency. Therefore, processing a work later than doing it early involves more machine time.

14*Corresponding author: *Mohamed.abdulfatah@uomustansiriyah.edu.iq*

Metallurgy, maintenance scheduling, and cleaning assignments are some typical examples of problems where the job processing time depends on the commencement time and increases with it. In these fields, any delay frequently indicates that more work (or time) will be required to complete the task. The concept of analyzing scheduling difficulties when a job's processing duration depends on when it starts was first described by Browne and Yechiali [4]. Miosheiov [5] assumed that the linear degrading model's fundamental processing times would not change when analyzing the flow time minimization problem. Researchers created models in which a job's real processing time is a function of its start time , Alidaee and Womer [6] divided scheduling models with degrading jobs into three categories: piecewise linear functions, non-linear functions, and linear functions.

Bachman and Janiak [7] shown that, under the linear degradation assumption, the maximum lateness minimization is NP-complete, and two heuristic algorithms were presented as consequence. Ng. et al. [8] reduce the overall completion time on a single machine, three scheduling issues with degrading workloads were examined, Ching and Ding [9] Considering the single machine problem to reduce the job lateness sum of squares using the basic linear degradation assumption, Chung [10] investigated the viability of scheduling a collection of start-time-dependent activities on a single computer with similar starting processing times, known deadlines, and processing rates.

On the other hand, Issues of the second kind are those where a non-increasing function determines the task processing time. These models have applications in machine maintenance, computer science, radar science, emergency medicine, and firefighting, among other fields. A non-increasing start time dependent function can be used to illustrate the so-called "learning effect." Let's say a worker has a lot of comparable things to assemble. The amount of time a worker needs to build a single product relies on a number of factors, including his knowledge, abilities, and workspace arrangement. The employee gains production skills. After some time, his knowledge has grown, his workspace is more orderly, and he is more skillful. Assembling a single product takes less time. The method by which a radar station is supposed to identify aerial threats is another example[11]. Here, a radar station has picked up several incoming objects. As the things approach closer, it takes less time to recognize them. Therefore, the less time required for object recognition, the later the items are recognized.

Ho et al.[11] established this approach after taking into account the issue of whether a solution is feasible given time restrictions. Examined were three scheduling problems using a decreasing linear model of the job processing times by Ng et al.[12] two of the problems have optimum solutions, and the goal function is to minimize the overall completion time. The linear models with rising and decreasing start time dependent components showed some interesting relationships. The single machine scheduling problem with start time dependent task processing times was examined by Bachman et al. They demonstrated that reducing the overall weighted completion time is an NP-hard task. Scheduling issues were examined by Wang and Xia[13] under a unique kind of linear decreasing degradation. In order to reduce makespan, maximum lateness, maximum cost, and number of late tasks, respectively, they proposed the best single machine scheduling methods. They demonstrated that Johnson's rule may be used to determine the ideal plan addressing the scheduling problem of the two device flow shop, which seeks to reduce the makespan. They showed how, if the processing times for each operation are the same for each task, the flow shop scheduling problem might be reduced to just one machine scheduling problem.

Dehua and Yunqiang [14] provides a thorough scheduling model that, by simultaneously taking into account the effects of learning and deterioration, greatly expands on a number of models that are already accessible in the literature. The impacts of learning and degradation, or the start timings and sequence locations, influence how long a job takes to process. lee and lai [15] provide a flexible scheduling strategy that leaves the function's shape unrestricted. The proposed model states that the anticipated location of the jobs and their past processing times are general functions of the actual job processing time.

The goal of this research is to reduce the overall completion time on a single machine by analyzing two scheduling issues with degrading workloads. A decreasing/increasing linear function of the work's execution start time determines the processing time for each job. There are no deadline or ready time constraints. In section 2, the problems are defined precisely. The primary objective of this research is to examine the distinctions between issues with linear models of task processing times that decrease and those that increase. This discrepancy stems from certain characteristics discovered for the two scheduling issues in both models. In section 3, it is a summary of the findings from this study. Some of the topics being researched are presented to the reader in the section 4.

2. PROBLEM DESCRIPTION

We create and characterize the issue we are considering, and there are ń jobs must be processing by single machine , all jobs ready at the time zero to be processed and the basic processing time ρ of job i ($i = 1,2, \dots, ń$) without interruption or preemption , the machine is available all the time and it can handle no more the one job on a time , but the actual processing time of the jobs grows (deteriorating) with its starting time , each job have different rate of growth λ_i , i.e. $P_i = \rho + \lambda_i t_i$.

In below some notations which we need in the remainder of the research

$P_1 = \rho$ (the actual processing of job 1 when the starting time $t_1 = 0$).

$P_2 = \rho + \lambda_2 P_1 = \rho + \lambda_2 \rho = \rho(1 + \lambda_2)$ (The starting time for job 2 is actually when the first job is done).

$P_3 = \rho + \lambda_3(P_2) = \rho + \lambda_3(\rho(1 + \lambda_2)) = \rho(1 + \lambda_3 + \lambda_3\lambda_2)$.

$P_4 = \rho + \lambda_4(P_3) = \rho + \lambda_4(\rho(1 + \lambda_3 + \lambda_3\lambda_2)) = \rho(1 + \lambda_4 + \lambda_4\lambda_3 + \lambda_4\lambda_3\lambda_2)$.

$P_i = \rho\, (1 + \lambda_i + \lambda_i\lambda_{i-1} + \lambda_i\lambda_{i-1}\lambda_{i-2} + \cdots + \cdots + \lambda_i\lambda_{i-1}\lambda_{i-2} \dots \lambda_1)$.

To calculate the completion time of job i :

$C_i = \sum_{l=1}^{i} P_l = P_1 + P_2 + P_3 + \cdots + P_i$.

$= \rho + \rho(1 + \lambda_2) + \rho(1 + \lambda_3 + \lambda_3\lambda_2) + \rho(1 + \lambda_4 + \lambda_4\lambda_3 + \lambda_4\lambda_3\lambda_2) + \cdots$

$+ \rho\, (1 + \lambda_i + \lambda_i\lambda_{i-1} + \lambda_i\lambda_{i-1}\lambda_{i-2} + \cdots + \cdots + \lambda_i\lambda_{i-1}\lambda_{i-2} \dots \lambda_1) = \rho \sum_{l=1}^{i} \prod_{ř=l+1}^{i} (1 + \lambda_ř)$.

For a given schedule $\pi = (1,2,3, \dots, ǰ, ǰ + 1, \dots, ń)$, $C_ǰ = C_ǰ(\pi)$ represent the completion time of the job ǰ, then we have ,

$$C_ǰ = \rho \sum_{l=1}^{ǰ} \prod_{ř=l+1}^{ǰ} (1 + \lambda_ř).$$

And,

$$C_{ǰ+1} = (1 + \lambda_{ǰ+1})C_ǰ + \rho$$

$$C_{ǰ+2} = (1 + \lambda_{ǰ+2})(1 + \lambda_{ǰ+1})C_ǰ + (1 + \lambda_{ǰ+2})\rho + \rho.$$

Theorem 2.1: The problem $1\left| P_i = \rho + \lambda_i t_i \right| \sum_{i=1}^{n} C_i$ has optimal objective function when the jobs scheduling by v-shaped with respect to the job increasing rate.

Proof. Let $\pi = (1,2,3, \dots, i - 1, i, 1 + 1, i + 2 , \dots, ń)$ be optimal with the following conditions

$\lambda_{\pi(i)} - \lambda_{\pi(i-1)} > 0$ ańd $\lambda_{\pi(i)} - \lambda_{\pi(i+1)} > 0$ (that means the v-shaped not satisfy).

And let π^* and π^{**} are two permutations get it by interchange the place of jobs i with $i - 1, i + 1$ at permutation π respectively.

Now,

$$\sum_{ǰ=1}^{ń} C_{\pi(ǰ)} - \sum_{ǰ=1}^{ń} C_{\pi^*(ǰ)}$$

$$= C_{\pi(1)} + C_{\pi(2)} + \cdots + C_{\pi(i-1)} + C_{\pi(i)} + C_{\pi(i+1)} + C_{\pi(i+2)} + \cdots + C_{\pi(ń)} - C_{\pi^*(1)} -$$

$$C_{\pi^*(2)} - \cdots - C_{\pi^*(i)} - C_{\pi^*(i-1)} - C_{\pi^*(i+1)} - C_{\pi^*(i+2)} - \cdots - C_{\pi^*(ń)} .$$

The completion time of any jobs at the permutations π and π^* its same before job $i - 2$,then we get,

$$\sum_{j=1}^{n} C_{\pi(j)} - \sum_{j=1}^{n} C_{\pi^*(j)} = (1+\lambda_{i-1})C_{i-2} + \rho + (1+\lambda_i)(1+\lambda_{i-1})C_{i-2} + (1+\lambda_i)\rho + \rho$$

$$+(1+\lambda_{i+1})(1+\lambda_i)(1+\lambda_{i-1})C_{i-2} + (1+\lambda_{i+1})(1+\lambda_i)\rho + (1+\lambda_{i+1})\rho + \rho + (1+\lambda_{i+2})(1+\lambda_{i+1})$$

$$(1+\lambda_i)(1+\lambda_{i-1})C_{i-2} + (1+\lambda_{i+2})(1+\lambda_{i+1})(1+\lambda_i)\rho + (1+\lambda_{i+2})(1+\lambda_{i+1})\rho +$$

$$(1+\lambda_{i+2})\rho + \rho + \cdots + (1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_i)(1+\lambda_{i-1})C_{i-2}$$

$$+(1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_i)\rho + (1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots$$

$$(1+\lambda_{i+1})\rho + \cdots + (1+\lambda_n)(1+\lambda_{n-1})\rho + (1+\lambda_n)\rho + \rho - (1+\lambda_i)C_{i-2} - \rho - (1+\lambda_{i-1})(1+\lambda_i)C_{i-2}$$

$$-(1+\lambda_{i-1})\rho - \rho - (1+\lambda_{i+1})(1+\lambda_{i-1})(1+\lambda_i)C_{i-2} - (1+\lambda_{i+1})(1+\lambda_{i-1})\rho - (1+\lambda_{i+1})\rho - \rho$$

$$-(1+\lambda_{i+2})(1+\lambda_{i+1})(1+\lambda_{i-1})(1+\lambda_i)C_{i-2} - (1+\lambda_{i+2})(1+\lambda_{i+1})(1+\lambda_{i-1})\rho - (1+\lambda_{i+2})(1+\lambda_{i+1})\rho$$

$$-(1+\lambda_{i+2})\rho - \rho - \cdots - (1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_{i-1})(1+\lambda_i)C_{i-2}$$

$$-(1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_{i-1})\rho - (1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})\rho$$

$$- \cdots - (1+\lambda_n)(1+\lambda_{n-1})\rho - (1+\lambda_n)\rho - \rho.$$

$$= (1+\lambda_{i-1})C_{i-2} + (1+\lambda_i)\rho + (1+\lambda_{i+1})(1+\lambda_i)\rho + (1+\lambda_{i+2})(1+\lambda_{i+1})(1+\lambda_i)\rho + \cdots$$

$$+(1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_i)\rho - (1+\lambda_i)C_{i-2} - (1+\lambda_{i-1})\rho.$$

$$-(1+\lambda_{i+1})(1+\lambda_{i-1})\rho - (1+\lambda_{i+2})(1+\lambda_{i+1})(1+\lambda_{i-1})\rho - \cdots - (1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1})(1+\lambda_{i-1})\rho.$$

$$= -(\lambda_i - \lambda_{i-1})C_{i-2} + (\lambda_i - \lambda_{i-1})\rho + (\lambda_i - \lambda_{i-1})\rho(1+\lambda_{i+1}) + (\lambda_i - \lambda_{i-1})\rho(1+\lambda_{i+2})(1+\lambda_{i+1}) + \cdots + (\lambda_i - \lambda_{i-1})\rho(1+\lambda_n)(1+\lambda_{n-1})(1+\lambda_{n-2}) \ldots (1+\lambda_{i+1}).$$

Then,

$$\sum_{j=1}^{n} C_{\pi(j)} - \sum_{j=1}^{n} C_{\pi^*(j)} = \rho(\lambda_i - \lambda_{i-1})\left[(1+\lambda_{i+1}) + (1+\lambda_{i+1})\sum_{j=i+2}^{n}\prod_{r=i+2}^{j}(1+\lambda_r) - \sum_{j=2}^{i-2}\prod_{r=2}^{i-2}(1+\lambda_r)\right] \quad (1)$$

By following the same way,

$$\sum_{\check{j}=1}^{\acute{n}} C_{\pi(\check{j})} - \sum_{\check{j}=1}^{\acute{n}} C_{\pi^{**}(\check{j})}$$
$$= \rho\left(\lambda_{\underline{i}} - \lambda_{\underline{i}+1}\right)\left[\left(1+\lambda_{\underline{i}-1}\right) + \left(1+\lambda_{\underline{i}-1}\right)\sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}}) - \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}})\right] \qquad (2)$$

By the optimality of π , both of (1) and (2) are non –positive, therefore

$$\left(1+\lambda_{\underline{i}+1}\right) + \left(1+\lambda_{\underline{i}+1}\right)\sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) - \sum_{\check{j}=2}^{\underline{i}-2}\prod_{\check{r}=2}^{\underline{i}-2}(1+\lambda_{\check{r}}) \le 0$$

$$\left(1+\lambda_{\underline{i}+1}\right) + \left(1+\lambda_{\underline{i}+1}\right)\sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) \le \sum_{\check{j}=2}^{\underline{i}-2}\prod_{\check{r}=2}^{\underline{i}-2}(1+\lambda_{\check{r}})$$

$$\left(1+\lambda_{\underline{i}+1}\right) + \lambda_{\underline{i}+1}\sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) \le \sum_{\check{j}=2}^{\underline{ii}-2}\prod_{\check{r}=2}^{\underline{i}-2}(1+\lambda_{\check{r}}) - \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}})$$

$$1+\lambda_{\underline{i}+1}\left[1 + \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}})\right]$$
$$\le \sum_{\check{j}=2}^{\underline{i}-2}\prod_{\check{r}=2}^{\underline{i}-2}(1+\lambda_{\check{r}}) - \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) \qquad (3)$$

And,

$$\left(1+\lambda_{\underline{i}-1}\right) + \left(1+\lambda_{\underline{i}-1}\right)\sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}}) - \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) \le 0$$

$$\left(1+\lambda_{\underline{i}-1}\right) + \left(1+\lambda_{\underline{i}-1}\right)\sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}}) \le \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}})$$

$$\left(1+\lambda_{\underline{i}-1}\right) + \lambda_{\underline{i}-1}\sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}}) \le \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) - \sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}})$$

$$1+\lambda_{\underline{i}-1}\left[1 + \sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}})\right] \le \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) - \sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}})$$

$$-1-\lambda_{\underline{i}-1}\left[1 + \sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}})\right]$$
$$\ge \sum_{\check{j}=1}^{\underline{i}-2}\prod_{\check{r}=\check{j}}^{\underline{i}-2}(1+\lambda_{\check{r}}) - \sum_{\check{j}=\underline{i}+2}^{\acute{n}}\prod_{\check{r}=\underline{i}+2}^{\check{j}}(1+\lambda_{\check{r}}) \qquad (4)$$

From (3) and (4),

$$-1-\lambda_{i-1}\left[1+\sum_{j=1}^{i-2}\prod_{r=j}^{i-2}(1+\lambda_r)\right]\geq 1+\lambda_{i+1}\left[1+\sum_{j=i+2}^{n}\prod_{r=i+2}^{j}(1+\lambda_r)\right]$$

$$0>-2\geq\lambda_{i+1}\left[1+\sum_{j=i+2}^{n}\prod_{r=i+2}^{j}(1+\lambda_r)\right]+\lambda_{i-1}\left[1+\sum_{j=1}^{i-2}\prod_{r=j}^{i-2}(1+\lambda_r)\right]>0$$

This is contradiction.

On the other hand, when the actual processing time of the jobs decreasing with its starting time, each job has different rate of decreasing $đ_i$, i.e. $P_i=\rho-đ_i t_i$ then,

$$\sum_{i=1}^{n}C_{\pi(i)}=\sum_{i}^{n}\rho\left[\sum_{j=1}^{i}\prod_{r=j+1}^{i}(1-đ_{\pi(r)})\right]$$

Where $đ_i$ denotes to the decreasing rate of job i , satisfy the following conditions:

$$0<đ_i<1 \ and \ đ_i<\frac{1}{n-1}.$$

For every unit delay in the job's beginning time, the first condition guarantees that the processing time drop is less than one unit. The second one guarantees that, within a workable timeline, all job processing times are positive, that leads to the next theorem.

Theorem 2.2: The problem $1\left|P_i=\rho-đ_i t_i\right|\sum_{i=1}^{n}C_i$ has optimal objective function when the jobs scheduling by Λ-shaped with respect to the job decreasing rates.

Proof. Let $\pi=(1,2,3,\dots,i-1,i,1+1,i+2,\dots,n)$ be optimal with the following conditions

$đ_{\pi(i-1)}-đ_{\pi(i)}>0 \ and \ đ_{\pi(i+1)}-đ_{\pi(i)}>0$ (that means the Λ-shaped not satisfy). And let π^* and π^{**} are two permutations get it by interchange the place of jobs i with $i-1,i+1$ at permutation π respectively.

Now,

$$\sum_{j=1}^{n}C_{\pi(j)}-\sum_{j=1}^{n}C_{\pi^*(j)}$$
$$=C_{\pi(1)}+C_{\pi(2)}+\dots+C_{\pi(i-1)}+C_{\pi(i)}+C_{\pi(i+1)}+C_{\pi(i+2)}+\dots+C_{\pi(n)}-C_{\pi^*(1)}-C_{\pi^*(2)}-\cdots$$
$$-C_{\pi^*(i)}-C_{\pi^*(i-1)}-C_{\pi^*(i+1)}-C_{\pi^*(i+2)}-\dots-C_{\pi^*(n)}.$$

The completion time of any jobs at the permutations π and π^* its same before job $i-2$.

then we get,

$$\sum_{j=1}^{n}C_{\pi(j)}-\sum_{j=1}^{n}C_{\pi^*(j)}$$
$$=(1-đ_{i-1})C_{i-2}+\rho+(1-đ_i)(1-đ_{i-1})C_{i-2}+(1-đ_i)\rho+\rho+(1-đ_{i+1})(1-đ_i)(1-đ_{i-1})C_{i-2}$$
$$+(1-đ_{i+1})(1-đ_i)\rho+(1-đ_{i+1})\rho+\rho+(1-đ_{i+2})(1-đ_{i+1})(1-đ_i)(1-đ_{i-1})C_{i-2}$$
$$+(1-đ_{i+2})(1-đ_{i+1})(1-đ_i)\rho+(1-đ_{i+2})(1-đ_{i+1})\rho+(1-đ_{i+2})\rho+\rho+\cdots$$
$$+(1-đ_n)(1-đ_{n-1})(1-đ_{n-2})\dots(1-đ_{i+1})(1-đ_i)(1-đ_{i-1})C_{i-2}$$
$$+(1-đ_n)(1-đ_{n-1})(1-đ_{n-2})\dots(1-đ_{i+1})(1-đ_i)\rho+(1-đ_n)(1-đ_{n-1})(1-đ_{n-2})\dots(1-đ_{i+1})\rho$$

$$+\cdots+(1-đ_{ń})(1-đ_{ń-1})\rho+\ (1-đ_{ń})\rho+\rho-\left(1-đ_{i}\right)C_{i-2}-\rho-\left(1-đ_{i-1}\right)\left(1-đ_{i}\right)C_{i-2}-\left(1-đ_{i-1}\right)\rho$$

$$-\rho-\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\left(1-đ_{i}\right)C_{i-2}-\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho-\left(1-đ_{i+1}\right)\rho-\rho$$

$$-\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\left(1-đ_{i}\right)C_{i-2}-\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho$$

$$-\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)\rho-\left(1-đ_{i+2}\right)\rho-\rho-\cdots-(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\left(1-đ_{i}\right)C_{i-2}$$

$$-(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho$$

$$-(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right)\rho-\cdots-(1-đ_{ń})(1-đ_{ń-1})\rho-(1-đ_{ń})\rho-\rho.$$

$$=\left(1-đ_{i-1}\right)C_{i-2}+\left(1-đ_{i}\right)\rho+\left(1-đ_{i+1}\right)\left(1-đ_{i}\right)\rho+\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)\left(1-đ_{i}\right)\rho+\cdots$$

$$+(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right)\left(1-đ_{i}\right)\rho-\left(1-đ_{i}\right)C_{i-2}-\left(1-đ_{i-1}\right)\rho$$

$$-\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho-\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho-\cdots$$

$$-(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right)\left(1-đ_{i-1}\right)\rho.$$

$$=-\left(đ_{i-1}-đ_{i}\right)C_{i-2}+\left(đ_{i-1}-đ_{i}\right)\rho+\left(đ_{i-1}-đ_{i}\right)\rho\left(1-đ_{i+1}\right)+\left(đ_{i-1}-đ_{i}\right)\rho\left(1-đ_{i+2}\right)\left(1-đ_{i+1}\right)$$

$$+\cdots+\left(đ_{i-1}-đ_{i}\right)\rho(1-đ_{ń})(1-đ_{ń-1})(1-đ_{ń-2})\ldots\ \left(1-đ_{i+1}\right).$$

$$=\rho\left(đ_{i-1}-đ_{i}\right)\left[1+\left(1-đ_{i+1}\right)+\left(1-đ_{i+1}\right)\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})-\sum_{ǰ=1}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})\right]$$

$$\sum_{ǰ=1}^{ń}C_{\pi(ǰ)}-\sum_{ǰ=1}^{ń}C_{\pi^{*}(ǰ)}=\rho\left(đ_{i-1}-đ_{i}\right)\left[\left(1-đ_{i+1}\right)+\left(1-đ_{i+1}\right)\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})-\sum_{ǰ=2}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})\right]\quad(1)$$

By follow the same way,

$$\sum_{ǰ=1}^{ń}C_{\pi(ǰ)}-\sum_{ǰ=1}^{ń}C_{\pi^{**}(ǰ)}=\rho\left(đ_{i+1}-đ_{i}\right)\left[\left(1-đ_{i-1}\right)+\left(1-đ_{i-1}\right)\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})-\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})\right]\quad(2)$$

From the optimality of π , both of (1) and (2) are non –positive, therefore,

$$\left(1-đ_{i+1}\right)+\left(1-đ_{i+1}\right)\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})-\sum_{ǰ=2}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})\ \leq 0$$

$$\left(1-đ_{i+1}\right)+\left(1-đ_{i+1}\right)\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) \le \sum_{ǰ=2}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})$$

$$\left(1-đ_{i+1}\right)-đ_{i+1}\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) \le \sum_{ǰ=2}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})-\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})$$

$$1-đ_{i+1}\left[1+\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})\right] \le \sum_{ǰ=2}^{i-2}\prod_{ř=2}^{i-2}(1-đ_{ř})-\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) \qquad (3)$$

And,

$$\left(1-đ_{i-1}\right)+\left(1-đ_{i-1}\right)\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})-\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) \le 0$$

$$\left(1-đ_{i-1}\right)+\left(1-đ_{i-1}\right)\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř}) \le \sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})-\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})$$

$$1-đ_{i-1}(1+\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř}) \le \sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})-\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})$$

$$-1+đ_{i-1}\left[1+\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})\right] \ge \sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})-\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) \qquad (4)$$

From (3) and (4),

$$1-đ_{i+1}\left[1+\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})\right] \le -1+đ_{i-1}\left[1+\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})\right]$$

$$2 \le đ_{i+1}\left[1+\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř})\right]+đ_{i-1}\left[1+\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř})\right]$$

Since $0 < đ_i < 1$,then,

$$\sum_{ǰ=2}^{i-2}\prod_{ř=ǰ}^{i-2}(1-đ_{ř}) < i-2$$

And,

$$\sum_{ǰ=i+2}^{ń}\prod_{ř=i+2}^{ǰ}(1-đ_{ř}) < ń-i-2 < ń-i-1.$$

Since $đ_i < \frac{1}{ń-1}$, then,

$$2 \leq đ_{i+1}\left[1+\sum_{j=i+2}^{ń}\prod_{ř=i+2}^{j}(1-đ_{ř})\right]+ đ_{i-1}\left[1+\sum_{j=2}^{i-2}\prod_{ř=j}^{i-2}(1-đ_{ř})\right].$$

$$2 \leq \frac{1}{ń-1}\ [1+ń-i-2]+\frac{1}{ń-1}\ [1+i-2].$$

$$2 \leq \frac{1}{ń-1}\ [1+ń-i-1]+\frac{1}{ń-1}\ [1+i-2].$$

$$2 \leq \frac{1}{ń-1}\ ń-1=1.$$

This is contradictory. Consequently, any optimal schedule must be Λ-shaped with respect to the decreasing rates.

The problem of uncertainty is very significant in production scheduling, just like it is in any other area of the manufacturing realm. The scheduling system's unpredictability may result from a variety of factors, including inconsistent processing times, malfunctioning machinery, and inaccurate client deadline information. Some negative consequences of not appropriately addressing or accounting for the unpredictability in the scheduling parameters involve surplus inventories and system instability, customer unhappiness due to missed deadlines, hence, effective scheduling under uncertainty has therefore become essential for businesses to achieve agreed shipment dates and make efficient use of the resources at their disposal in the age of agile and lean production. Therefore, coming up with the appropriate scheduling techniques to use in practice is crucial.

The problem of project scheduling in particular or unclear circumstances has been studied by scholars since the 1960s. The first person to apply probability theory to the project scheduling issue was Freeman [16]. Using programming with chance constraints, Cooper and Charnes [17] examined the problem of stochastic project scheduling. McKay , Safayeni and Buzacott [18] contends that scheduling theory's inability to sufficiently account for significant variations in processing durations are the primary cause of its poor applicability in real-world scenarios. Dodin [19] claims that when all jobs are arranged to take their projected durations deterministically, the pseudo-deterministic sequence that results from doing so does not accurately reflect the goals of stochastic analysis of a schedule. Additionally, he recommends using a different sequence that was selected using an Optimality Indices (OI) rating method. OI is the probability that each sequence has of being the best. Mittenthal and Raghavachari [20] On a single machine that is prone to unplanned failures, they present the scheduling task of minimizing an anticipated sum of no regular penalty functions. A simple recourse model is studied and a deterministic equivalent goal function is created where the punishment function is the squared divergence of task completion times from a common due date. The deviation function shows the predicted sum of squares of the difference between the tasks' total required time and the previously established due date time. where the objective is to minimize the squared deviation of job completion times from a due date, the challenge is to identify a sequence of weighted sums of squared deviation (WSSD) of the completion of job about a common due date is minimize (and it SSD when the jobs have equal weights), take a look at this issue, the due date is constant or every job has a different due date. The problem is describing by:

$$D = E\left[\sum_{i=1}^{ń} w_i (C_i - d_i)^2\right].$$

Where w_i is weighted of job i and assume the due date d_i are common / different for each jobs $i = 1,2,3,\dots,ń$ which are independent in the sequence, refer to this problem as $1/\,Ƥ_i = \rho + ƛ_i t_i / E\left[\sum_{i=1}^{ń}(C_i - d_i)^2\right]$.

We consider, the case where the processing time is random variable with the expected and variance, then we can calculate the expected of completion times as,

$$E(C_j) = E\left(\rho \sum_{l=1}^{j} \prod_{ř=l+1}^{j}(1+ƛ_{ř})\right) = \sum_{l=1}^{j} \prod_{ř=l+1}^{j}(1+ƛ_{ř})\, E(\rho).$$

And the variance of completion times as

$$Var(C_{j}) = Var\left(\rho \sum_{l=1}^{j} \prod_{r=l+1}^{j} (1 + \lambda_{r})\right) = \sum_{l=1}^{j} \left(\prod_{r=l+1}^{j} (1 + \lambda_{r})\right)^{2} Var\ (\rho).$$

Corollary 2.3: The problem $1\left| P_{i} = \rho + \lambda_{i} t_{i} \right| E\left[\sum_{i=1}^{n} C_{i}\right]$ has optimal objective function when the jobs scheduling by (v-shaped) with respect to the job increasing rate.

Corollary 2.4: The problem $1\left| P_{i} = \rho + \lambda_{i} t_{i} \right| Var\left[\sum_{i=1}^{n} C_{i}\right]$ has optimal objective function when the jobs scheduling by (v-shaped) with respect to the job increasing rate.

Then, we can form the deviation problem as

$$D = E\left[\sum_{i=1}^{n} w_{i}(C_{i} - d_{i})^{2}\right] = E\left[\sum_{i=1}^{n} w_{i}(C_{i}^{2} - 2C_{i}d_{i} + d_{i}^{2})\right] = \sum_{i=1}^{n} w_{i}\left[E(C_{i}^{2}) - 2d_{i}E(C_{i}) + d_{i}^{2}\right]$$

$D = \sum_{i=1}^{n} w_{i}\left[Var(C_{i}) + (E(C_{i}))^{2} - 2d_{i}E(C_{i}) + d_{i}^{2}\right]$, that will lead us to the next theorem.

Theorem 2.5: The deviation problem $1\left| P_{i} = \rho + \lambda_{i} t_{i} \right| E\left[\sum_{i=1}^{n} w_{i}(C_{i} - d_{i})^{2}\right]$ has optimal objective function when the jobs scheduling by (v-shaped) with respect to the job increasing rate .

Proof. From the inequality,

$$\left(E(C_{i}) - d_{i}\right)^{2} = \left(E(C_{i})\right)^{2} - 2d_{i}E(C_{i}) + d_{i}^{2}$$

$$\left(E(C_{i}) - d_{i}\right)^{2} + 2d_{i}E(C_{i}) = \left(E(C_{i})\right)^{2} + d_{i}^{2}$$

$$2d_{i}E(C_{i}) \leq \left(E(C_{i})\right)^{2} + d_{i}^{2}$$

where $\sum_{i=1}^{n} d_{i}$ is independent about sequence, by corollary 2.3, 2.4 then,

$$D = \sum_{i=1}^{n} w_{i}\left[Var(C_{i}) + (E(C_{i}))^{2} - 2d_{i}E(C_{i}) + d_{i}^{2}\right].$$

has optimal objective function when the jobs scheduling by (v-shaped) with respect to the job increasing rate.

Corollary 2.6: The problem $1\left| P_{i} = \rho - d_{i} t_{i} \right| E\left[\sum_{i=1}^{n} C_{i}\right]$ has optimal objective function when the jobs scheduling by (Λ-shaped) with respect to the job decreasing rates.

Corollary 2.7: The problem $1\left| P_{i} = \rho - d_{i} t_{i} \right| Var\left[\sum_{i=1}^{n} C_{i}\right]$ has optimal objective function when the jobs scheduling by (Λ-shaped) with respect to the job decreasing rates.

Theorem 2.8: The deviation problem $1\left| P_{i} = \rho - d_{i} t_{i} \right| E\left[\sum_{i=1}^{n} w_{i}(C_{i} - d_{i})^{2}\right]$ has optimal objective function when the jobs scheduling by (Λ-shaped) with respect to the job decreasing rates.

Proof. Same technique in theorem 2.5.

3. CONCLUSIONS AND OPEN PROBLEM

The scheduling research community has focused more of its attention on scheduling issues related to the impacts of the recent linear degradation. In this study, given that the starting window period is a constant basic processing time, a single machine-scheduling model with a decreasing and increasing linear degradation is introduced. We have demonstrated that, in the case when the implementation time is constant, the only scheduling difficulties may still be solved over if the goals are to minimize overall total completion times ($1| Ƥ_i = \rho - đ_i t_i | \sum_{i=1}^{n} C_i$) ,($1| Ƥ_i = \rho + ƛ_i t_i | \sum_{i=1}^{n} C_i$) and a deviation problem ($1| Ƥ_i = \rho - đ_i t_i | E\left[\sum_{i=1}^{ń} w_i (C_i - d_i)^2\right]$), ($1| Ƥ_i = \rho + ƛ_i t_i | E\left[\sum_{i=1}^{ń} w_i (C_i - d_i)^2\right]$) using a version of Λ-shaped/ v-shaped linear deterioration. We think that scale theory and applications will benefit more from the model that is being provided here. To help practitioners select the appropriate scheduling guidelines, it would be helpful and the appropriate model in practical cases.

The deviation problem when $Ƥ_i = \rho_i + ƛ_i t_i$ or $Ƥ_i = \rho_i - ƛ_i t_i$ (the basic process is not constant) it is thought to be NP-hard, but a proof is still pending, and which does not yield to a simple analysis even for exceptional circumstances, where the processing time is a random. Therefore, if the processing time is in a state of linear deterioration and the goal function is of the multiple type, we suggest a set of scheduling issues on a single machine, , see [21] which are as follows;

$1| Ƥ_i = \rho_i - đ_i t_i | \sum_{i=1}^{ń} (C_i + T_i + E_i)$, $1| Ƥ_i = \rho_i - đ_i t_i | \sum_{i=1}^{ń} (C_i + T_i + E_i + V_i)$, $1| Ƥ_i = \rho_i - đ_i t_i | \sum_{i=1}^{ń} (C_i + T_{max} + E_{max})$,

The objective is to find the optimal solutions using the branch-and-bound method, dominance rules can be derived for these functions that help us in the branch-and-bound method, approximate solutions can also be found using the local search algorithms method.

REFERENCE

[1] F. H. Ali and M. G. Ahmed, "Local Search Methods for Solving Total Completion Times, Range of Lateness and Maximum Tardiness Problem," in *Proceedings of the 6th International Engineering Conference "'Sustainable Technology and Development'", IEC 2020*, IEEE, Feb. 2020, pp. 103–108. doi: 10.1109/IEC49899.2020.9122821.

[2] H. A. Chachan, F. H. Ali, and M. H. Ibrahim, "Branch and Bound and Heuristic Methods for Solving Multi-Objective Function for Machine Scheduling Problems," in *2020 6th International Engineering Conference "Sustainable Technology and Development" (IEC)*, IEEE, Feb. 2020, pp. 109–114. doi: 10.1109/IEC49899.2020.9122793.

[3] M. H. Ibrahim, F. H. Ali, and H. A. Chachan, "Dominance Rules for Single Machine Scheduling to Minimize Weighted Multi Objective Functions," in *AIP Conference Proceedings*, Feb. 2023, p. 040090. doi: 10.1063/5.0119047.

[4] S. Browne and U. Yechiali, "Scheduling deteriorating jobs on a single processor," *Oper Res*, vol. 38, no. 3, pp. 495–498, Jun. 1990, doi: 10.1287/opre.38.3.495.

[5] G. Mosheiov, "V-shaped policies for scheduling deteriorating jobs," *Oper Res*, vol. 39, no. 6, pp. 979–991, Dec. 1991, doi: 10.1287/opre.39.6.979.

[6] B. Alidaee and N. K. Womer, "Scheduling with time dependent processing times: Review and extensions," *J Oper Res Soc*, vol. 50, no. 7, pp. 711–720, Jul. 1999, doi: 10.1057/palgrave.jors.2600740.

[7] A. Bachman and A. Janiak, "Minimizing maximum lateness under linear deterioration," *Eur J Oper Res*, vol. 126, no. 3, pp. 557–566, Nov. 2000, doi: 10.1016/S0377-2217(99)00310-0.

[8] C. T. Ng, T. C. E. Cheng, A. Bachman, and A. Janiak, "Three scheduling problems with deteriorating jobs to minimize the total completion time," *Inf Process Lett*, vol. 81, no. 6, pp. 327–333, Mar. 2002, doi: 10.1016/S0020-0190(01)00244-7.

[9] T. C. E. Cheng and Q. Ding, “Scheduling start time dependent tasks with deadlines and identical initial processing times on a single machine,” *Comput Oper Res*, vol. 30, no. 1, pp. 51–62, Jan. 2003, doi: 10.1016/S0305-0548(01)00077-6.

[10] Y.-H. Chung, H.-C. Liu, C.-C. Wu, and W.-C. Lee, “A deteriorating jobs problem with quadratic function of job lateness,” *Comput Ind Eng*, vol. 57, no. 4, pp. 1182–1186, Nov. 2009, doi: 10.1016/j.cie.2009.05.012.

[11] K. I. J. Ho, J. Y. T. Leung, and W. D. Wei, “Complexity of scheduling tasks with time-dependent execution times,” *Inf Process Lett*, vol. 48, no. 6, pp. 315–320, Dec. 1993, doi: 10.1016/0020-0190(93)90175-9.

[12] A. Bachman, T. C. E. Cheng, A. Janiak, and C. T. Ng, “Scheduling start time dependent jobs to minimize the total weighted completion time,” *J Oper Res Soc*, vol. 53, no. 6, pp. 688–693, Jun. 2002, doi: 10.1057/palgrave.jors.2601359.

[13] J. B. Wang and Z. Q. Xia, “Scheduling jobs under decreasing linear deterioration,” *Inf Process Lett*, vol. 94, no. 2, pp. 63–69, Apr. 2005, doi: 10.1016/j.ipl.2004.12.018.

[14] Y. Yin and D. Xu, “Some single-machine scheduling problems with general effects of learning and deterioration,” *Comput Math with Appl*, vol. 61, no. 1, pp. 100–108, 2011, doi: 10.1016/j.camwa.2010.10.036.

[15] W.-C. Lee and P.-J. Lai, “Scheduling problems with general effects of deterioration and learning,” *Inf Sci (Ny)*, vol. 181, no. 6, pp. 1164–1170, Mar. 2011, doi: 10.1016/j.ins.2010.11.026.

[16] R. J. Freeman, “Letter to the Editor—A Generalized PERT,” *Oper Res*, vol. 8, no. 2, pp. 281–281, Apr. 1960, doi: 10.1287/opre.8.2.281.

[17] A. Charnes and W. W. Cooper, “Chance-Constrained Programming,” *Manage Sci*, vol. 6, no. 1, pp. 73–79, Oct. 1959, doi: 10.1287/mnsc.6.1.73.

[18] K. N. McKay, F. R. Safayeni, and J. A. Buzacott, “Job-Shop Scheduling Theory: What Is Relevant?,” *Interfaces (Providence)*, vol. 18, no. 4, pp. 84–90, Aug. 1988, doi: 10.1287/inte.18.4.84.

[19] B. Dodin, “Determining the optimal sequences and the distributional properties of their completion times in stochastic flow shops,” *Comput Oper Res*, vol. 23, no. 9, pp. 829–843, Sep. 1996, doi: 10.1016/0305-0548(95)00083-6.

[20] J. Mittenthal and M. Raghavachari, “Stochastic Single Machine Scheduling with Quadratic Early-Tardy Penalties,” *Oper Res*, vol. 41, no. 4, pp. 786–796, Aug. 1993, doi: 10.1287/opre.41.4.786.

[21] H. A. Cheachan and M. T. Kadhim, “Exact Method for Solving Single Machine Scheduling Problem under Fuzzy Due Date to Minimize Multi-Objective Functions,” *J Phys Conf Ser*, vol. 1879, no. 2, May 2021, doi: 10.1088/1742-6596/1879/2/022126.