CHAPTER-II

MULTI-JOBS IN SINGLE MACHINE SCHEDULING PROBLEM WITH NON-LINEAR DETERIORATED AND TIME-DEPENDENT LEARNING

The main results of this chapter have been published as detailed below:

Multi-Jobs in Single Machine Scheduling Problem with Non-Linear Deteriorated and Time-Dependent Learning.

International Journal of Statics and System (IJSA) Volume 2, Number 1, PP. 27-34, 2012.

2.1 INTRODUCTION

We study the multi-jobs single machine scheduling problem under the effect of nonlinear job deterioration and time-dependent. The single machine scheduling problem with non-linear deterioration learning and present the complexity results concerning time dependent scheduling on a single machine. Also present the results concerning a single machine and minimization of the $C_{\max}$, $\sum C_j$ and $L_{\max}$ criterion. We assume that the processing time of a job increases when it's processing is delayed. The objectives are considered: the makespan, the sum of completion times (square) and the minimum lateness.

A task j is a fundamental entity described in time domain by a start time and finish time that is executed for a processing time on a certain resource. The scheduling problems are very important in the fields of manufacturing systems. The multi-jobs in single machine, tardiness problems may be stated as follows: a set of n jobs (numbered 1, 2… n) with known processing time $p_1, p_2, ..., p_n$ and due dates $d_1, d_2, ..., d_n$ are to be processed without interruption on multi-jobs in single machine. All jobs available at time zero and set up time of the jobs are included in the job-processing times and independent of the processing sequence.

Alidaceet et al. [1] consider the nonlinear deterioration effect proposed by given in the following equation,

$$P_{[r]} = P_r + \alpha t_r^b \quad (2.1.1)$$

where $P_{[r]}$ and t_r are the actual processing time and the starting time of job that is processed in the r^{th} position, respectively. P_r is basic processing time of the job. α (α>0) and b (b>0) are parameters of the nonlinear deterioration effects, which determines the increase in the processing time of a job per unit delay in its starting time. **Kuo et al.**[2] considered the situation where the learning curve is a time dependent i.e. the processing time of a job not only depend on its starting time, but also depend on a total normal processing time of the already processing jobs. **Biskup** [3] discussed some of the economic fundamentals of scheduling and learning effects, and presented an extensive survey of different scheduling models and problems involving jobs. **Chenget et al.** [4] considered some scheduling problem with deteriorating jobs and learning effects i.e. the actual processing time of a job dependents not only on the processing of the jobs already processed but also on its scheduled position. They showed that the single machine problems are polynomially solvable if the performance criterion is makespan, total completion time, total weighted completion time or maximum lateness. **J. B. Wang** [5] discussed the single machine scheduling problems with a time dependent learning effect and simple linear deterioration. He proved that the single machine make span minimization problem remain polynomially solvable. **Xiao-Yuan et al.** [6] considered several single machine problems under the simultaneous effect of nonlinear deterioration and time-dependent learning. This is the most general model studied to date.

The present chapter deals with to develop a framework for better deodorizing real life system where the rate of deteriorated increases or decreases over time and where the learning is driven by time rather than by the number of completed tasks, such a scenario can arise in many realistic situations.

2.2 MINIMUM TOTAL COMPLETION TIME

In this section, we study a single-machine scheduling problem with a time-dependent learning effect. For a given schedule q, let $C_i = C_i(q)$ represent the completion time of J_i. The total completion time is denoted by $\sum_{i=1}^{n} C_i$. For convenience, we denote the time-dependent learning effect mentioned by LE_t . Thus, using the conventional notation, the problem of total completion time minimization on a single machine is denoted by $1/LE_t/\sum_{i=1}^{n} C_i$.

The classical scheduling problem of minimizing the total completion time on a single machine is optimized by the SPT policy. In addition, **Biskup** (1999) showed that the total completion time minimization problem with a job independent learning effect ($1/LE_t/\sum_{i=1}^{n} C_i$) was also optimized by sequencing jobs according to the SPT rule. In this section, we show that when a time-dependent learning effect is assumed for each job, $1/LE_t/\sum_{i=1}^{n} C_i$ is still optimized by the SPT rule.

2.3 FORMULATION OF THE PROBLEM

The notation and terminology given in **Toksari et al.** [7]. There are n jobs $J_1, J_2, ..., J_n$ to be processed on a single machine each of them is available at time zero. The machine can handle one job at a time and preemption is not allowed.

If job J_i is scheduled in the r^{th} position of the processing sequence, then its actual processing time $P_{[r]}$ is defined as.

$$P_{[r]} = [P_r + (\alpha \times t_r^b)](1 + \sum_{k=1}^{r-1} p_{[r]})^a \tag{2.3.1}$$

Where $\sum_{k=1}^{r-1} p_{[r]} = 0$, $P_{[r]}$ is the basic processing time of Job J_i, α (α>0) and b (b>0) are parameters of the nonlinear deterioration effects, which determines the increase in the processing time of a job per unit delay in its starting time. a (a ≤ 0) is the learning index , and t_r is starting time of the job scheduling in passion r. **Toksari et al.** [7] give the following results for the single machine problem.

MAKESPAN MINIMIZATION

In this section, we study a single-machine group scheduling problem with a time-dependent learning effect. The objective is to minimize the makespan of all jobs. For convenience, we denote the time dependent learning effect mentioned LE_t . In addition, let G denote that the problem is a group scheduling problem and S denote the existence of a sequence-independent group setup time. Therefore, using the conventional notation, the single-machine group scheduling problem with a time-dependent learning effect and the sequence independent group setup time is denoted by $1/G,S,LE_t/C_{\max}$

2.4 SOME SINGLE MACHINE SCHEDULING PROBLEMS

A practical example that motivates the above scheduling model is the manual production of glass crafts by a skilled craftsman. Silicon-based raw material is first heated up in an oven until it becomes a lump of malleable dough from which the craftsman cuts pieces and shapes them according to different designs into different glass craft products. The initial time to heat up the raw material to the threshold temperature at which it can be shaped is long and so the first piece has a long processing time, which includes both the heating time (i.e., the deterioration effect) and the shaping time (i.e., the normal processing time). The second piece requires a shorter time to re-heat the dough to the threshold temperature (i.e., a smaller deterioration effect). Similarly, the later a piece is cut from the dough, the shorter is its heating time to reach the threshold temperature. On the other hand, the pieces that are shaped later require shorter shaping times because the craftsman's productivity improves as a result of learning.

Let $C_{\max} = \max\{C_j j = 1,2,...,n\}, \sum C_j, \sum C_j^2$

and $L_{\max} = \max\{C_j j = 1,2,...,n\}$

represent the makespan, the sum of completion times, and the sum of completion time square and maximum lateness of a given permutation, respectively.

First, we prove the following lemma:

Lemma 1.

$$\Big[\big((\lambda \times A) - A\big) + \Big(A \times \big((\lambda \times t) + 1\big)^a\Big) - \Big((\lambda \times A) \times (t+1)^a\Big) + \Big(\alpha \times (t+1)^a$$

$$\times \Big(\big(A \times \lambda \times x^a\big) + y\Big)^b\Big) - \Big(\alpha \times \big((\lambda + t) + 1\big)^a \times \Big(\big(A \times x^a\big) + y\Big)^b\Big)\Big] \geq 0$$

when

$(A > 0)(a < 0), (\alpha > 0), (\lambda > 0), (t \geq 0)\, and\, (\lambda \geq 1)$.

Proof. $f(\lambda) = \Big[\big((\lambda \times A) - A\big) + \Big(A \times \big((\lambda \times t) + 1\big)^a\Big) - \Big((\lambda \times A) \times (t+1)^a\Big)$

$$+\Big(\alpha \times (t+1)^a \times \Big(\big(A \times \lambda \times x^a\big) + y\Big)^b\Big)$$

$$-\Big(\alpha \times \big((\lambda + t) + 1\big)^a \times \Big(\big(A \times x^a\big) + y\Big)^b\Big)\Big] \qquad (2.4.1)$$

We know that if the derivate $f'(x)$ of a continuous function $f(x)$ satisfies $f'(x) > 0$ on an interval (a,b), then $f(x)$ is increasing on

(a,b). In this case, by taking the first and second derivatives of Eq. (2.4.1) with respect to λ, we obtain:

$$f'(\lambda)=\Big[A+\left(A\times a\times t\times\left((\lambda\times t)+1\right)^{a-1}\right)-\left(A\times(t+1)^{a}\right)$$
$$+\left(\alpha\times(t+1)^{a}\times A\times x^{a}\times b\times\left(\left(A\times\lambda\times x^{a}\right)+y\right)^{b-1}\right)$$
$$-\left(\alpha\times\left(\left(A\times x^{a}\right)+y\right)^{b}\times a\times t\times\left((\lambda\times t)+1\right)^{a-1}\right)\Big]$$

and

$$f''(\lambda)=\Big[\left(A\times a\times(a-1)\times t^{2}\times\left((\lambda\times t)+1\right)^{a-2}\right)$$
$$+\left(\alpha\times(t+1)^{a}\times A^{2}\times x^{2a}\times b\times(b-1)\times\left(\left(A\times\lambda\times x^{a}\right)+y\right)^{b-2}\right)$$
$$-\left(\alpha\times\left(\left(A\times x^{a}\right)+y\right)^{b}\times a\times t^{2}\times(a-1)\times\left((\lambda\times t)+1\right)^{a-2}\right)\Big]$$

It implies that $f''(\lambda)\geq 0$. Therefore, $f'(\lambda)$ is non-decreasing function for $a<0,\alpha,b,A>0,t\geq 0\,and\,\lambda\geq 1$.

$$\Big[\left((\lambda\times A)-A\right)+\left(A\times\left((\lambda\times t)+1\right)^{a}\right)-\left((\lambda\times A)\times(t+1)^{a}\right)+\left(\alpha\times(t+1)^{a}\right.$$
$$\left.\times\left(\left(A\times\lambda\times x^{a}\right)+y\right)^{b}\right)-\left(\alpha\times\left((\lambda+t)+1\right)^{a}\times\left(\left(A\times x^{a}\right)+y\right)^{b}\right)\Big]\geq 0$$

for $a<0,\alpha,b,A>0,t\geq 0\,and\,\lambda\geq 1$.

Theorem 1. *The makespan problem on a single machine under the effects of nonlinear deterioration and time-dependent learning*

$$1\left|\left[p_r+\left(\alpha\times t_r^b\right)\right]\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right|C_{\max}$$

can be solved optimally by sequencing jobs in non-decreasing order of their basic processing times.

Proof. Consider an optimal schedule π, which contains two adjacent jobs, job J_u followed by job $J_u\left(v=u+1\right)$, such that $p_u<p_v$. The starting time of J_u is T and C_u and C_v express the completion time of the jobs scheduled in the position $u\left(J_u\right)$ and v and $\left(J_u\right)$, respectively $\left(v=u+1\right)$, with the nonlinear effects of learning and deterioration we obtain:

$$C_v\left(\pi\right)=T+\left(\left(p_u+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)$$
$$+\left(\left(p_v+\left(\alpha\times\left(\left(p_u+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}+p_u\right)^a\right).$$

By performing a pairwise interchange on jobs J_u and J_v, we obtain schedule π' where the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v are not affected by interchange, and thus,

$$C_v(\pi') = T + \left(\left(p_v + \left(\alpha \times T^b \right) \right) \left(1 + \sum_{k=1}^{r-1} p_{[k]} \right)^a \right)$$
$$+ \left(p_u + \left(\left(\alpha \times \left(\left(p_v + \left(\alpha \times T^b \right) \right) \left(1 + \sum_{k=1}^{r-1} p_{[k]} \right)^a \right)^b \right) \right) \left(1 + \sum_{k=1}^{r-1} p_{[k]} + p_v \right) \right).$$

By substituting $x = 1 + \sum_{k=1}^{r-1} p_{[k]}$ we obtain the difference the difference between the values of $C_v(\pi)$ and $C_u(\pi')$:

$$C_u(\pi') - C_v(\pi) = \left(x^a \times (p_v - p_u) \right) + \left(p_u \times (p_v + x)^a \right) - \left(p_v \times (p_u + x)^a \right)$$
$$+ \left(\alpha \times (p_u \times x)^a \times \left(\left(p_v \times x^a \right) + \left(\alpha \times T^b \times x^a \right) \right)^b \right)$$
$$- \left(\alpha \times (p_v \times x)^a \times \left(\left(p_u \times x^a \right) + \left(\alpha \times T^b \times x^a \right) \right)^b \right). \qquad (2.4.2)$$

substituting $y = \left(\alpha \times T^b \times x^a \right)$, $t = \dfrac{p_u}{x}$ *and* $\lambda = \dfrac{p_v}{p_u}$ we obtain:

$$\frac{C_u(\pi') - C_v(\pi)}{x^a} = \left((\lambda \times p_u) - p_u \right) + \left(p_u \times \left((\lambda \times t) + 1 \right)^a \right)$$
$$- \left((\lambda \times p_u) \times (t+1)^a \right) - \left((\lambda \times p_u) \times (t+1)^a \right) + \left(\alpha \times (t+1)^a \right.$$
$$\left. \times \left(\left(p_u \times \lambda \times x^a \right) + y \right)^b \right) - \left(\alpha \times \left((\lambda \times t) + 1 \right)^a \times \left(\left(p_u \times x^a \right) + y \right)^b \right). \qquad (2.4.3)$$

From lemma 1, it follows that:

$$\frac{C_u(\pi') - C_v(\pi)}{x^a} = \left[\left((\lambda \times p_u) - p_u \right) + \left(p_u \times \left((\lambda \times t) + 1 \right)^a \right) - \left((\lambda \times p_u) \times (t+1)^a \right) \right.$$

$$+\left(\alpha\times(t+1)^{a}\times\left(\left(p_{u}\times\lambda\times x^{a}\right)+y\right)^{b}\right)$$

$$-\left(\alpha\times\left((\lambda\times t)+1\right)^{a}\times\left(\left(p_{u}\times x^{a}\right)+y\right)^{b}\right)\Bigg]\geq 0.$$

Consequently, $C_{u}(\pi')>C_{v}(\pi)$.

The makespan under π is strictly smaller then that under π'. This contradicts the optimality of π' and completes the proof.

Theorem 2. *The total completion time problem on a single machine under the effects of nonlinear deterioration and time-dependent learning* $1\left|\left[p_{r}+\left(\alpha\times t_{r}^{b}\right)\right]\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^{a}\right|Cj$, *can be solved optimally by sequencing jobs in non-decreasing order of their basic processing times.*

Proof. Consider an optimal schedule π, which contains two adjacent jobs, job J_{u} followed by job $J_{v}\,(v=u+1)$, such that $p_{u}<p_{v}$. T is total completion time of all job before J_{u} when the starting time of J_{u} is T and C_{u} and C_{v} express the completion time of J_{u} and J_{v}, the jobs scheduled at position u and $(v=u+1)$, respectively.

$$C_{v}(\pi)=T+\left(2\times\left(p_{u}+\left(\alpha\times T^{b}\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^{a}\right)$$

$$+\left(\left(p_{v}+\left(\alpha\times\left(\left(p_{u}+\left(\alpha\times T^{b}\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^{a}\right)^{b}\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}+p_{u}\right)^{a}\right).$$

By interchange on jobs J_u and J_v, we obtain schedule π' where the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v are not affected by interchange, and thus,

$$C_v(\pi') = T + \left(2\times\left(p_v + \left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1} p_{[k]}\right)^a\right)$$
$$+\left(p_u + \left(\left(\alpha\times\left(\left(p_v+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1} p_{[k]}\right)^a\right)^b\right)\right)\left(1+\sum_{k=1}^{r-1} p_{[k]} + p_v\right)\right).$$

Again substituting $x = 1+\sum_{k=1}^{r-1} p_{[k]}$ yields that the difference between the values of $C_v(\pi)$ and $C_u(\pi')$ can be expressed as follows:

$$C_u(\pi') - C_v(\pi) = \left(2\times x^a \times (p_v - p_u)\right) + \left(p_u\times(p_v+x)^a\right) - \left(p_v\times(p_u+x)^a\right)$$
$$+\left(\alpha\times(p_u\times x)^a\times\left(\left(p_v\times x^a\right)+\left(\alpha\times T^b\times x^a\right)\right)^b\right)$$
$$-\left(\alpha\times(p_v\times x)^a\times\left(\left(p_u\times x^a\right)+\left(\alpha\times T^b\times x^a\right)\right)^b\right). \qquad (2.4.4)$$

By substituting $y = \left(\alpha\times T^b\times x^a\right),\ t = \frac{p_u}{x}\ and\ \lambda = \frac{p_v}{p_u}$ we obtain:

$$\frac{C_u(\pi') - C_v(\pi)}{x^a} = \left(2\times\left((\lambda\times p_u) - p_u\right)\right) + \left(p_u\times\left((\lambda\times t)+1\right)^a\right)$$

$$-\left((\lambda\times p_u)\times(t+1)^a\right) + \alpha\times(t+1)^a\times\left(\left(p_u\times\lambda\times x^a\right)+y\right)^b\Big)$$

$$-\left(\alpha\times\left((\lambda\times t)+1\right)^a\times\left(\left(p_u\times x^a\right)+y\right)^b\right). \tag{2.4.5}$$

From lemma 1, it follows that:

$$\left[\frac{C_u(\pi') - C_v(\pi)}{x^a} = \left((\lambda\times p_u) - p_u\right) + \left(p_u\times\left((\lambda\times t)+1\right)^a\right)\right.$$

$$-\left((\lambda\times p_u)\times(t+1)^a\right) + \left(\alpha\times(t+1)^a\times\left(\left(p_u\times\lambda\times x^a\right)+y\right)^b\right)$$

$$\left.-\left(\alpha\times\left((\lambda\times t)+1\right)^a\times\left(\left(p_u\times x^a\right)+y\right)^b\right)\right] \geq 0.$$

Implying that $\sum C(\pi') > \sum C(\pi)$.

π Clearly dominates π', contradicting the optimality of π'.

Theorem 3. *The total completion time (square) problem on a single machine under the effects of nonlinear deterioration and time-dependent learning,* $1\left|\left[p_r + \left(\alpha\times t_r^b\right)\right]\left(1+\sum_{k=1}^{r-1} p_{[k]}\right)^a\right|C_j^2$, *can be solved* *optimally by sequencing jobs in non-decreasing order of their basic processing times.*

Proof. Follows directly forms Theorem 2. Since $C_u(\pi') > C_v(\pi)$ and $C_v(\pi') > C_u(\pi)$, it clearly follows that $C_u^2(\pi') > C_v^2(\pi)$ and $C_v^2(\pi') > C_u^2(\pi)$.

Theorem 4. *The maximum lateness problem on a single machine under the effects of nonlinear deterioration and time-dependent learning,* $1\left|\left[p_r + \left(\alpha \times t_r^b\right)\right]\left(1+\sum_{k=1}^{r-1} p_{[k]}\right)^a\right|L_{\max}$, *if jobs have agreeable due dates can be solved optimally by sequencing jobs in non-decreasing order of their due dates* d_j.

Proof. Consider an optimal schedule π, which contains two adjacent jobs, job J_u followed by job $J_v\,(v = u+1)$, such that $p_u < p_v$. T The starting time of J_u is T and L_u and L_v express the lateness of J_u and J_v, the jobs scheduled at position u and $(v = u+1)$, respectively. The lateness of this job pair is:

$$L_u(\pi) = T + \left(\left(p_u + \left(\alpha \times T^b\right)\right)\left(1+\sum_{k=1}^{r-1} p_{[k]}\right)^a\right) - d_u$$

$$L_v(\pi)=\Bigg[T+\left(p_u+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a$$

$$+\left(\left(p_v+\left(\alpha\times\left(\left(p_u+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}+p_u\right)^a\right)\Bigg]-d_v$$

$$L_u(\pi)=C_u(\pi)-d_u.$$

Performing a pairwise interchange on jobs J_u and J_v, we obtain schedule π' where the starting time of J_v is T. The completion times of the jobs processed before jobs J_u and J_v are not affected by interchange, and the lateness of the jobs pair is now:

$$L_v(\pi')=T+\left(\left(p_v+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)-d_u$$

$$L_u(\pi')=\Bigg[T+\left(\left(p_v+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)$$

$$+\left(p_u+\left(\left(\left(\alpha\times\left(\left(p_v+\left(\alpha\times T^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}\right)^a\right)^b\right)\right)\left(1+\sum_{k=1}^{r-1}p_{[k]}+p_v\right)\right)\right)\Bigg]-d_u$$

$$L_u(\pi')=C_u(\pi')-d_u.$$

The difference between the values of $L_v(\pi')$ *and* $L_u(\pi)$ is

$$L_v(\pi') - L_u(\pi) = \left((p_v - p_u)\left(1 + \sum_{k=1}^{r-1} p_{[k]}\right)^a \right) + (d_u - d_v). \tag{2.4.6}$$

Furthermore, the difference between the values of $L_u(\pi')$ and $L_v(\pi)$ is

$$L_u(\pi') - L_v(\pi) = (C_u(\pi') - C_v(\pi)) + (d_u - d_v). \tag{2.4.7}$$

It follows form Theorem 1 that $C_u(\pi') > C_v(\pi)$ when $p_u < p_v$, $a < 0$ and $b < 0$. In addition, $L_v(\pi') - L_u(\pi) < 0$ and $L_u(\pi') - L_v(\pi) > 0$ are obtained using Eqs. (2.4.6) and (2.4.7) when $d_u < d_v$. π dominates π', which contradicts the optimality of π'. This completes the proof.

2.5 COUNTER EXAMPLE

In the following example, we show that the results of theorems 1-4 are correct by giving a counter-example.

Example1. Let $n = 5$, $p_1 = 8$, $p_2 = 12$, $p_3 = 20$, $p_4 = 25$ and $p_5 = 32$; $d_i = 5$(i=1, 2,…,5); $\alpha = 0.1$ $a = -5, b = 0.1$ if the jobs are scheduled to be processed according to the SPT rule, and the sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4) , then according to the result of theorem 1-4,

Table 1.1(a):

Using the sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4) then result

Sr.No.	Basic Processing time	Processing time of job i (p_i)	Completion time of job i (c_i)	Square c_i^2
1	8	8	8	64
3	32	0.0000015601	8.000353	64.0056481
4	12	0.000000152	8.0003530152	64.0056483
5	25	0.0000000124	8.0003530276	64.0056485

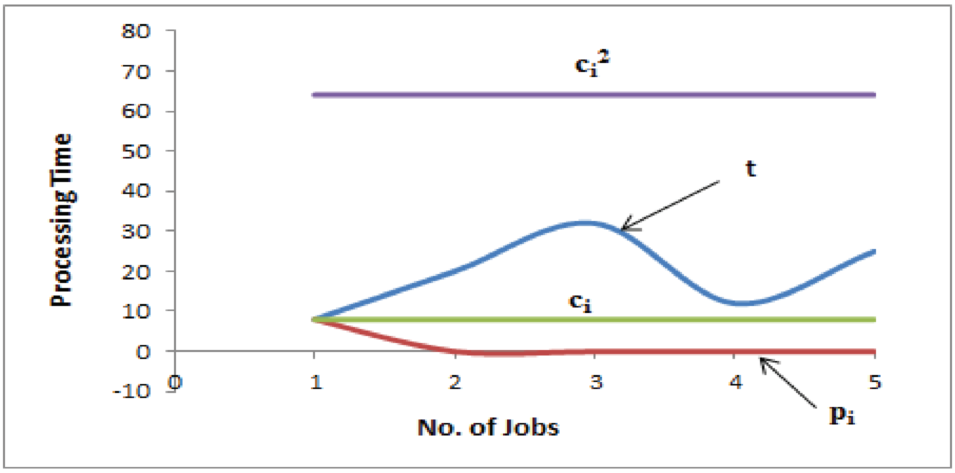

Figure 1.1(a): *Sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4).*

Table 1.1(b): Using the SPT Rule

Sr.No.	Basic Processing time	Processing time of job i (p_i)	Completion time of job i (c_i)	Square c_i^2
1	8	8	8	64
2	12	0.0002167	8.000216	64.003456047
3	20	0.0000050929	8.0002210	64.003536049
4	25	0.0000002227	8.000221227	64.003539249
5	32	0.0000000262	8.0002212532	64.003540049

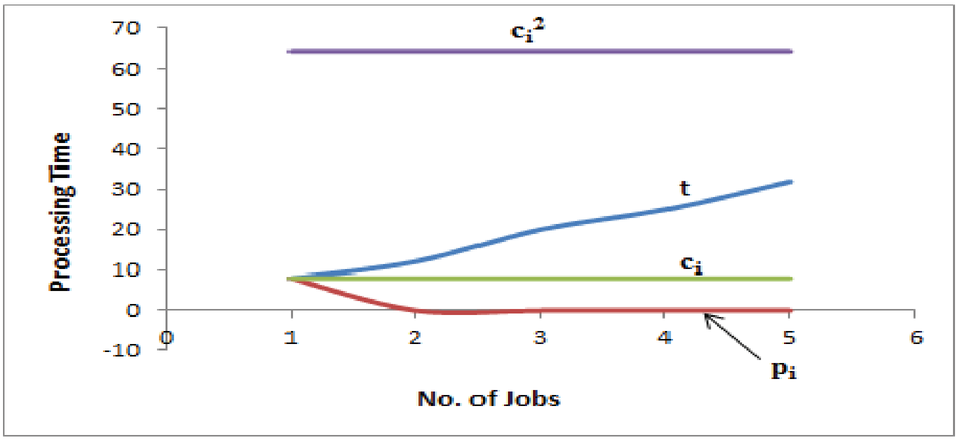

Figure 1.1(b): *The SPT Rule*

Example2. Let $n=4, p_1=2, p_2=5, p_3=8,$ and $p_4=10; d_i=4$ (i=1, 2, 3, 4); α = 20 $a=-2, b=1$ if the jobs are scheduled to be processed according to the SPT rule, and the sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4), then according to the result of theorem 1-4.

Table 2.1(a): The sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4)

Sr.No.	Basic Processing time	Processing time of job i (p_i)	Completion time of job i (c_i)	Square c_i^2
1	2	2	2	4
2	10	5.555	7.555	57.0780
3	5	0.9230	8.473	71.7917
4	8	0.5477	9.0207	81.3730

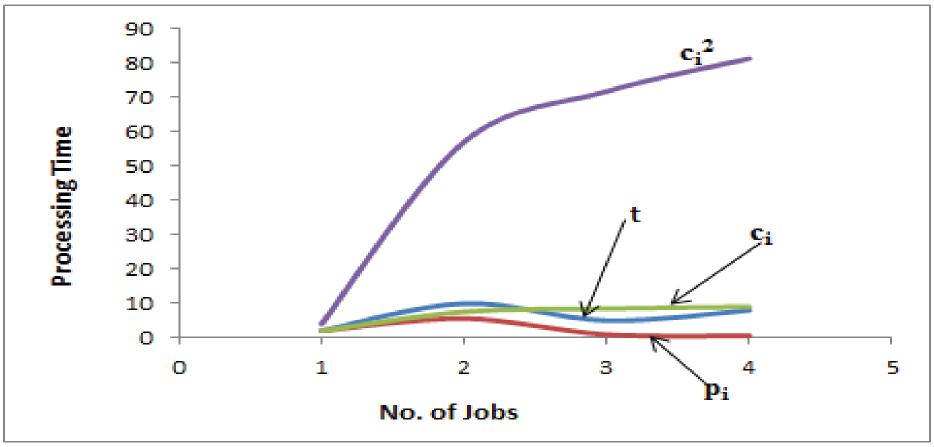

Figure 2.1(a): *Sequence of the jobs is (using J_1, J_3, J_5, J_2, J_4)*

Table 2.1(b): Using the SPT Rule

Sr.No.	Basic Processing time	Processing time of job i (p_i)	Completion time of job i (c_i)	Square c_i^2
1	2	2	2	4
2	5	5	7	49
3	8	2.3125	9.3125	86.7226
4	10	0.7666	10.0791	101.5882

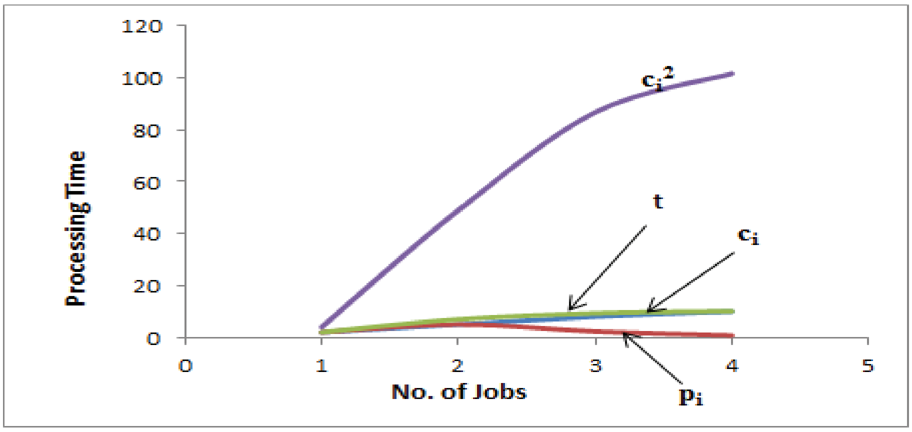

Figure 2.1(b): *The SPT Rule*

2.6 CONCLUSION

This chapter provided new deteriorating jobs under the proposed deteriorating jobs, we show that the multi-jobs in single machine makespan and basic processing time, processing time of job i, completion times of job i, square (c_i^2) are solvable. In addition, we studied using the theorem 1-4 proposed by Taskari et al. solve 1.1(a) and 2.1(a) tables while graphs are using the sequence of the jobs 1.1(b) and 2.1(b) table and also graphs are using the SPT rule of the jobs they obtained results for sequence and are showing graphically. The objective are minimize the makespan and the total completion time of all jobs.

2.7 REFERENCES

1. **Alidace, B., and Womer, N. K.:** Scheduling with Time Dependent Processing Time; Review and extensions, *the Journal of Operational Research Society* 50, (7) pp. 711-720, 1999.
2. **Kuo, W. H., and Yong, D. L.:** Minimizing the Total Completion Time in a Single-machine Scheduling Problem with Time-dependent Learning Effect, *Eur. J. Operation Research*, 174, pp. 1184-1190, 2006.
3. **Biskup, D.:** A State-of-art Review on Scheduling with Learning Effects, *Eur. J. Operation Research,* 188, pp. 315-319, 2008.
4. **Cheng, T. C. E., and Lec-C.:** Some Scheduling Problems with Deteriorating Jobs and Learning Effects, *Comp. Ind. Eng.* 54 pp. 972-982, 2008.
5. **Wang, J. B.:** Single Machine Scheduling with a Time-dependent Learning Effect and Deteriorating Jobs, *Operation Research, Soc*. 60, pp. 583-586, 2009.
6. **Xiao-Yuan, Wang., Li-Bo, Wong., and Erratum.:** Single Machine Scheduling Problems Under the Effects of Nonlinear Deterioration and Time-dependent Learning, *Learning Mathematical and Computer Modeling,*50, pp. 401-406, 2009.
7. **Tokseri, M. D., Oron, D., and Guner, E.:** Single Machine Scheduling Problems Under the Effects of Nonlinear Deterioration and Time-dependent Learning, *Mathematical and Computer Modeling*, 50, pp. 401-406, 2009.