ABSTRACT
This paper deals branch and bound technique to solve single machine scheduling problem involving two processing times along with due date using Type-2 Trapezoidal fuzzy numbers. Our aim is to obtain optimal sequence of jobs and to minimize the total tardiness. The working of the algorithm has been illustrated by numerical example.

AMS Subject Classification: 94D05, 90-XX, Aug 2010.

KEYWORDS: Branch and Bound Technique, Processing times P1 and P2, Optimal sequence, Type-2 Trapezoidal fuzzy numbers.

INTRODUCTION
Job scheduling is a useful tool in decision making problem. The scheduling problems are common occurrence in our daily life. The aim of this technique is used to determine an optimal job scheduling problem and minimizing the total tardiness. For many years, Scheduling problem focused on single performance measure.

In this paper, we propose a new concept in single machine scheduling problem. Recent development of new technology, we are consider the single machine having double processor to do two different works to complete a job. Each work having separate processing times (ie) P1 and P2 addition to the due date (dj). The most obvious objective is to scheduling the job and minimizing the total tardiness using Branch and Bound technique. This method is basically a stage wise search method of optimization problems whose solutions may be viewed as the result of a sequence of decisions that will help the decision maker in determining a best schedule for a given set of jobs effectively. This method is become lucrative to make decision. In most of the real life problem, there are elements of uncertainty in process. In practical situation processing times and due date are not always deterministic. So, we have associated with fuzzy environment.

The concept of a type-2 fuzzy set, which is an extension of the concept of an ordinary fuzzy set, was introduced by Zadeh [16]. A fuzzy set is two dimensional and a type-2 fuzzy set is three dimensional, type-2 fuzzy sets can better improve certain kinds of inference than do fuzzy sets with increasing imprecision, uncertainty and fuzziness in information. A type-2 fuzzy set is characterized by a membership function, (ie) the membership value for each element of this set is a fuzzy set in [0,1], unlike an ordinary fuzzy set where the membership value is a crisp number in [0,1].

REVIEW OF LITERATURE
bound method to solve the total earliness and total tardiness problem for the single machine problem. Raymond [14] proposed a branch and bound approach to solve the problem for steel plant involving single machine bi-criteria problem.

The paper is organized as follows: In section 2, deals with the preliminaries. In section 3, arithmetic operations on type-2 trapezoidal fuzzy number and ranking function are discussed. In section 4, we introduced a brief note on Branch and Bound Technique. In section 5, the effectiveness of the proposed method is illustrated by means of an example.

PRELIMINARIES
Definition: Fuzzy Set
A fuzzy set is characterized by a membership function mapping the elements of domain, space or universe of discourse X to the unit interval [0,1].

A fuzzy set $\tilde{A}$ is set of ordered pairs { (x, $\mu_{\tilde{A}}(x)$) / x $\in$ R} where $\mu_{\tilde{A}}(x) : R \rightarrow [0,1]$ is upper semi continuous function, $\mu_{\tilde{A}}(x)$ is called a membership function of the fuzzy set.

Definition: Fuzzy Number
A fuzzy number f in the real line R is a fuzzy set f: R $\rightarrow$ [0,1] that satisfies the following properties.
(i) f is piecewise continuous.
(ii) There exists an x $\in$ R such that f(x) =1.
(iii) f is convex (i.e) if $x_1, x_2$ $\in$ R and then $\lambda$ $\in$ [0,1] then 

$\mu_f(\lambda x_1 + (1-\lambda) x_2) \geq \mu_f(x_1) \land \mu_f(x_2)$.

Definition: Type-2 Fuzzy Set
The type-2 fuzzy sets are defined by functions of the form $\mu_{\tilde{A}} : X \rightarrow \tilde{\lambda}([0,1])$ where $\tilde{\lambda}([0,1])$ denotes the set of all ordinary fuzzy sets that can be defined within the universal set [0,1]. An example of a membership function of this type is given in fig-1.

Definition: Type-2 Fuzzy Number
Let $\tilde{A}$ be a type-2 fuzzy set defined in the universe of discourse R. If the following conditions are satisfied.
(i) $\tilde{A}$ is normal.
(ii) \( \tilde{A} \) is a convex set.

(iii) The support of \( \tilde{A} \) is closed and bounded, then \( \tilde{A} \) is called a type-2 fuzzy number.

**Definition: Trapezoidal Fuzzy Number**

A trapezoidal fuzzy number \( A = (a_1, a_2, a_3, a_4) \) whose membership function is given by

\[
\mu_A(x) = \begin{cases} 
0 & , x < a_1 \ & x > a_4 \\
\frac{x - a_1}{a_2 - a_1} & , a_1 \leq x \leq a_2 \\
1 & , a_2 \leq x \leq a_3 \\
\frac{a_4 - x}{a_4 - a_3} & , a_3 \leq x \leq a_4 
\end{cases}
\]

**Definition: Type-2 Trapezoidal Fuzzy Number**

A type-2 trapezoidal fuzzy number \( \tilde{A} \) on \( R \) is given by

\[
\tilde{A} = (\tilde{\mu}_{\tilde{A}_1}(x), \tilde{\mu}_{\tilde{A}_2}(x), \tilde{\mu}_{\tilde{A}_3}(x), \tilde{\mu}_{\tilde{A}_4}(x)), \quad x \in R
\]

and

\[
\mu_{\tilde{A}_1}(x) \leq \mu_{\tilde{A}_2}(x) \leq \mu_{\tilde{A}_3}(x) \leq \mu_{\tilde{A}_4}(x) \text{ for all } x \in R. \text{ (ie) } \tilde{A} = (\tilde{A}_1, \tilde{A}_2, \tilde{A}_3, \tilde{A}_4)
\]

\[
\tilde{A} = (a_{1L}, a_{1M}, a_{1N}, a_{1U}), (a_{2L}, a_{2M}, a_{2N}, a_{2U}), (a_{3L}, a_{3M}, a_{3N}, a_{3U}), (a_{4L}, a_{4M}, a_{4N}, a_{4U})
\]

**ARITHMETIC OPERATIONS**

**Arithmetic Operations on Type-2 Trapezoidal Fuzzy Numbers:**

Let \( \tilde{A} \) and \( \tilde{B} \) be two type-2 trapezoidal fuzzy numbers. Then, we define

**Addition:**

\[
\tilde{A} + \tilde{B} = ((a_{1L} + b_{1L}, a_{1M} + b_{1M}, a_{1N} + b_{1N}, a_{1U} + b_{1U}),(a_{2L} + b_{2L}, a_{2M} + b_{2M}, a_{2N} + b_{2N}, a_{2U} + b_{2U}),(a_{3L} + b_{3L}, a_{3M} + b_{3M}, a_{3N} + b_{3N}, a_{3U} + b_{3U}),(a_{4L} + b_{4L}, a_{4M} + b_{4M}, a_{4N} + b_{4N}, a_{4U} + b_{4U})).
\]

**Subtraction:**

\[
\tilde{A} - \tilde{B} = ((a_{1L} - b_{1U}, a_{1M} - b_{1N}, a_{1N} - b_{1L}, a_{1U} - b_{1L}),(a_{2L} - b_{2L}, a_{2M} - b_{2N}, a_{2N} - b_{2N}, a_{2U} - b_{2U}),(a_{3L} - b_{3U}, a_{3M} - b_{3N}, a_{3N} - b_{3N}, a_{3U} - b_{3U}),(a_{4L} - b_{4U}, a_{4M} - b_{4N}, a_{4N} - b_{4N}, a_{4U} - b_{4U})).
\]

**Multiplication:**

\[
\tilde{A} \times \tilde{B} = ((a_{1L} \times b_{1L}, a_{1M} \times b_{1M}, a_{1N} \times b_{1N}, a_{1U} \times b_{1U}),(a_{2L} \times b_{2L}, a_{2M} \times b_{2M}, a_{2N} \times b_{2N}, a_{2U} \times b_{2U}),(a_{3L} \times b_{3L}, a_{3M} \times b_{3M}, a_{3N} \times b_{3N}, a_{3U} \times b_{3U}),(a_{4L} \times b_{4L}, a_{4M} \times b_{4M}, a_{4N} \times b_{4N}, a_{4U} \times b_{4U})).
\]

**Division:**

\[
\tilde{A} / \tilde{B} = ((a_{1L} / b_{1L}, a_{1M} / b_{1M}, a_{1N} / b_{1N}, a_{1U} / b_{1U}),(a_{2L} / b_{2L}, a_{2M} / b_{2M}, a_{2N} / b_{2N}, a_{2U} / b_{2U}),(a_{3L} / b_{3L}, a_{3M} / b_{3M}, a_{3N} / b_{3N}, a_{3U} / b_{3U}),(a_{4L} / b_{4L}, a_{4M} / b_{4M}, a_{4N} / b_{4N}, a_{4U} / b_{4U})).
\]
Ranking on Type-2 Trapezoidal Fuzzy Number

Let \( F(\sigma) \) be the set of all type-2 normal trapezoidal fuzzy numbers. One convenient approach for solving numerical valued problems is based on the concept of comparison of fuzzy numbers by use of ranking function. An effective approach for ordering the elements of \( F(\sigma) \) is to define a linear ranking function \( R: F(\sigma) \rightarrow \sigma \) which maps each fuzzy number in \( \sigma \).

Suppose \( \tilde{A} = (\tilde{A}_1, \tilde{A}_2, \tilde{A}_3, \tilde{A}_4) \)

\[
\tilde{A} = \left(\begin{array}{cccc}
a_1^L & a_1^M & a_1^N & a_1^U \\
b_1^L & b_1^M & b_1^N & b_1^U \\
c_1^L & c_1^M & c_1^N & c_1^U \\
d_1^L & d_1^M & d_1^N & d_1^U \\
\end{array}\right)
\]

(\(a_1^L, a_1^M, a_1^N, a_1^U\), (\(a_2^L, a_2^M, a_2^N, a_2^U\)), (\(a_3^L, a_3^M, a_3^N, a_3^U\)), (\(a_4^L, a_4^M, a_4^N, a_4^U\))).

Then we define

\[
R(\tilde{A}) = a_1^L + a_1^M + a_1^N + a_1^U + a_2^L + a_2^M + a_2^N + a_2^U + a_3^L + a_3^M + a_3^N + a_3^U + a_4^L + a_4^M + a_4^N + a_4^U / 16.
\]

Also, we define orders on \( F(\sigma) \) by

\[
\begin{align*}
R(\tilde{A}) &\geq R(\tilde{B}) \text{ if and only if } \tilde{A} \geq \tilde{B} \\
R(\tilde{A}) &\leq R(\tilde{B}) \text{ if and only if } \tilde{A} \leq \tilde{B} \\
R(\tilde{A}) &= R(\tilde{B}) \text{ if and only if } \tilde{A} = \tilde{B}
\end{align*}
\]

BRANCH AND BOUND TECHNIQUE

Branch and Bound: Branching is the process of partitioning a large problem into two or more subproblems and Bounding is the process of calculating a lower bound on the optimal solution of a given subproblems.

Dominance Property: While subdividing a subproblem \( P_{\sigma}^k \) into (n-k) subproblems, a careful analysis would help us to create only one subproblem instead of n-k subproblems. This is called dominance property. This will reduce the computational effort to a greater extent. In a subproblem \( P_{\sigma}^k \), if there exists a job \( i \notin \sigma^l \) such that \( d_i \geq q_\sigma \), then it is sufficient to create only one subproblem \( P_{\sigma}^{k+1} \). The remaining subproblems under \( P_{\sigma}^k \) can be ignored. In the bounding process, \( V_{\sigma} = V_\sigma \).

Tardiness: Tardiness is the lateness of job \( j \) if it fails to meet its due date; otherwise, it is zero. It is defined as:

\[
T_j = \max \{ c_j - d_j \} = \max \{ 0, t_j \}
\]

Notations:

\[
\begin{align*}
n & : \text{ The total number of independent jobs.} \\
j & : \text{ Represents the } j^{th} \text{ job, } j = 1, 2, \ldots, N. \\
t_j & : \text{ The Processing time of the job } j. \\
d_j & : \text{ The due date of the job } j. \\
C_j & : \text{ The Completion time of the job } j. \\
T_j & : \text{ The Tardiness of the job } j. \\
NT_j & : \text{ Number of the tardy jobs.} \\
T_m & : \text{ Minimum tardiness.}
\end{align*}
\]
Algorithm

The processing times of jobs and due date are uncertain. This leads to the use of Type-2 trapezoidal fuzzy numbers for representing these imprecise values.

Step-I:
Place $P_0$ on the active list; its associated values are: $V_0 = 0$ and $q_0 = \sum_{j=1}^{n} t_j$. At a given stage of the algorithm, the active list consists of all the terminal nodes of the partial tree created up to that stage.

Step-2:
Remove the first subproblem $P_{\sigma_k}^k$ from the active list. If $k$ is equal to $n-1$, stop. Prefix the missing job with $\sigma$ and treat it as the optimal sequence. Otherwise, check the dominance property for $P_{\sigma_k}^k$. If the property holds, go to step 3; otherwise go to step 4.

Step-3:
Let the job $j$ be the job with the largest due date in $\sigma^1$. Create the subproblem $P_{\sigma_j^1}^{k+1}$ with $q_{\sigma_j^1} = q_\sigma - t_j$, $V_{\sigma_j^1} = V_{\sigma^1}$, $b_{\sigma_j^1} = V_\sigma$. Place $P_{\sigma_j^1}^{k+1}$ on the active list, ranked by its lower bound. Return to step 2.

Step-4:
Create $(n-k)$ subproblems, one for each $i \in \sigma^1$. For $P_{\sigma_i}^{k+1}$, let, $q_{\sigma_i} = q_\sigma - t_j$, $V_{\sigma_i} = V_0 + \max(0, q_\sigma - d_i)$, $b_{\sigma_i} = V_{\sigma_i}$. Now place each $P_{\sigma_i}^{k+1}$ on the active list, ranked by its lower bound. Return to step 2.

NUMERICAL ILLUSTRATION

In milk producing factory, they required double processor for production using single machine. There are two processes done by the single machine. (i) Crushed the soya to produce milk is the first process made by the machine. (ii) That milk will be packed by respective quantities is the second process made by the same machine. These two processes are having separate processing times ($P_1$, $P_2$). Here, we consider the two processing time and due date with type-2 trapezoidal fuzzy numbers for each jobs are given in the following table:

<table>
<thead>
<tr>
<th>Job $j$</th>
<th>Processing time $P_1$</th>
<th>Processing time $P_2$</th>
<th>Due date $d_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(5,6,10,11)</td>
<td>(4,6,8,10)</td>
<td>(14,18,22,26)</td>
</tr>
<tr>
<td></td>
<td>(4,6,10,12)</td>
<td>(3,6,8,11)</td>
<td>(13,18,22,27)</td>
</tr>
<tr>
<td></td>
<td>(3,6,10,13)</td>
<td>(2,6,8,12)</td>
<td>(12,18,22,28)</td>
</tr>
<tr>
<td></td>
<td>(2,6,10,14)</td>
<td>(1,6,8,13)</td>
<td>(11,18,22,29)</td>
</tr>
<tr>
<td>2</td>
<td>(1,3,5,7)</td>
<td>(1,3,5,7)</td>
<td>(12,14,16,18)</td>
</tr>
<tr>
<td></td>
<td>(0,3,5,8)</td>
<td>(0,3,5,8)</td>
<td>(11,14,16,19)</td>
</tr>
<tr>
<td></td>
<td>(-1,3,5,9)</td>
<td>(-1,3,5,9)</td>
<td>(10,14,16,20)</td>
</tr>
<tr>
<td></td>
<td>(-2,3,5,10)</td>
<td>(-2,3,5,10)</td>
<td>(9,14,16,21)</td>
</tr>
<tr>
<td>3</td>
<td>(5,8,10,13)</td>
<td>(4,7,9,12)</td>
<td>(12,24,36,48)</td>
</tr>
<tr>
<td>Job j</td>
<td>Processing time $P_1$</td>
<td>Processing time $P_2$</td>
<td>Processing time $t_j$</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>1</td>
<td>(5,6,10,11)</td>
<td>(4,6,8,10)</td>
<td>(9, 12,18,21)</td>
</tr>
<tr>
<td></td>
<td>(4,6,10,12)</td>
<td>(3,6,8,11)</td>
<td>(7,12,18,23)</td>
</tr>
<tr>
<td></td>
<td>(3,6,10,13)</td>
<td>(2,6,8,12)</td>
<td>(5,12,18,25)</td>
</tr>
<tr>
<td></td>
<td>(2,6,10,14)</td>
<td>(1,6,8,13)</td>
<td>(3,12,18,27)</td>
</tr>
<tr>
<td>2</td>
<td>(1,3,5,7)</td>
<td>(1,3,5,7)</td>
<td>(2,6,10,14)</td>
</tr>
<tr>
<td></td>
<td>(0,3,5,8)</td>
<td>(0,3,5,8)</td>
<td>(0,6,10,16)</td>
</tr>
<tr>
<td></td>
<td>(-1,3,5,9)</td>
<td>(-1,3,5,9)</td>
<td>(-2,6,10,18)</td>
</tr>
<tr>
<td></td>
<td>(-2,3,5,10)</td>
<td>(-2,3,5,10)</td>
<td>(-4,6,10,20)</td>
</tr>
<tr>
<td>3</td>
<td>(5,8,10,13)</td>
<td>(4,7,9,12)</td>
<td>(9,15,19,25)</td>
</tr>
<tr>
<td></td>
<td>(4,8,10,14)</td>
<td>(3,7,9,13)</td>
<td>(7,15,19,27)</td>
</tr>
<tr>
<td></td>
<td>(3,8,10,15)</td>
<td>(2,7,9,14)</td>
<td>(5,15,19,29)</td>
</tr>
<tr>
<td></td>
<td>(2,8,10,16)</td>
<td>(1,7,9,15)</td>
<td>(3,15,19,31)</td>
</tr>
<tr>
<td>4</td>
<td>(2,4,6,8)</td>
<td>(1,3,5,7)</td>
<td>(3,7,11,15)</td>
</tr>
<tr>
<td></td>
<td>(1,4,6,9)</td>
<td>(0,3,5,8)</td>
<td>(1,7,11,17)</td>
</tr>
<tr>
<td></td>
<td>(0,4,6,10)</td>
<td>(-1,3,5,9)</td>
<td>(-1,7,11,19)</td>
</tr>
<tr>
<td></td>
<td>(-1,4,6,11)</td>
<td>(-2,3,5,10)</td>
<td>(-3,7,11,21)</td>
</tr>
<tr>
<td>5</td>
<td>(2,5,7,10)</td>
<td>(2,5,7,10)</td>
<td>(4,10,14,20)</td>
</tr>
<tr>
<td></td>
<td>(1,5,7,11)</td>
<td>(1,5,7,11)</td>
<td>(2,10,14,22)</td>
</tr>
<tr>
<td></td>
<td>(0,5,7,12)</td>
<td>(0,5,7,12)</td>
<td>(0,10,14,24)</td>
</tr>
<tr>
<td></td>
<td>(-1,5,7,13)</td>
<td>(-1,5,7,13)</td>
<td>(-2,10,14,26)</td>
</tr>
</tbody>
</table>

**Step-1:**

Active list at level $0 = \{P^0 \phi\}, \sigma = \{\phi\}, \sigma' = \{1,2,3,4,5\}, V_\phi = 0$ & $q_\phi = \begin{bmatrix} 27, 50, 72, 95 \\ 17, 50, 72, 105 \\ 75, 72, 115 \\ -3, 50, 72, 125 \end{bmatrix}$. Since, the current level $k(0)$ is not equal to $n-1(4)$. Check the dominance property. Also $\max_{i \in \sigma} d_i = \begin{cases} 12, 24, 36, 48 \\ 11, 24, 36, 49 \\ 10, 24, 36, 50 \\ 9, 24, 36, 51 \end{cases}$. Since, this maximum is not greater than $q_\phi$. The details of computations of the lower bound for each of the node is
<table>
<thead>
<tr>
<th>$P_i^\sigma$</th>
<th>$V_{in} = V_\sigma + \max (0, q_{i\sigma} \cdot d_i)$</th>
<th>$b_{in} = v_{in}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1^1$</td>
<td>$0 + \max 0, [27, 50, 72, 95] - 17, 50, 72, 105 - 7, 50, 72, 115 - 3, 50, 72, 125$</td>
<td>$1, 28, 54, 81$</td>
</tr>
<tr>
<td>$P_2^1$</td>
<td>$0 + \max 0, [27, 50, 72, 95] - 17, 50, 72, 105 - 7, 50, 72, 115 - 3, 50, 72, 125$</td>
<td>$9, 34, 58, 83$</td>
</tr>
<tr>
<td>$P_3^1$</td>
<td>$0 + \max 0, [27, 50, 72, 95] - 17, 50, 72, 105 - 7, 50, 72, 115 - 3, 50, 72, 125$</td>
<td>$-21, 14, 48, 83$</td>
</tr>
<tr>
<td>$P_4^1$</td>
<td>$0 + \max 0, [27, 50, 72, 95] - 17, 50, 72, 105 - 7, 50, 72, 115 - 3, 50, 72, 125$</td>
<td>$-2, 29, 59, 90$</td>
</tr>
<tr>
<td>$P_5^1$</td>
<td>$0 + \max 0, [27, 50, 72, 95] - 17, 50, 72, 105 - 7, 50, 72, 115 - 3, 50, 72, 125$</td>
<td>$-13, 20, 52, 85$</td>
</tr>
</tbody>
</table>

Active list = \{ $P_1^1, P_3^1, P_1^4, P_4^1, P_5^1$ \}

**Step-2:**
Check the dominance property, $\sigma' = \{3\}, \sigma = \{1,2,4,5\}$.

\[ q_{\sigma'} = \begin{pmatrix}
27, 50, 72, 95 \\
17, 50, 72, 105 \\
7, 50, 72, 115 \\
-3, 50, 72, 125
\end{pmatrix} = \begin{pmatrix}
9, 15, 19, 25 \\
17, 15, 19, 27 \\
7, 5, 19, 29 \\
3, 15, 19, 31
\end{pmatrix} = \begin{pmatrix}
2, 31, 57, 86 \\
10, 31, 57, 98 \\
-22, 31, 57, 110 \\
-34, 31, 57, 122
\end{pmatrix}, \]

\[ \max_i d_i = \begin{pmatrix}
10, 20, 30, 40 \\
9, 20, 30, 41 \\
8, 20, 30, 42 \\
7, 20, 30, 43
\end{pmatrix}. \] Since, this maximum value is not greater than
The details of computations of the lower bound for each of the node is

\[
q_n = \begin{cases}
10,20,30,40 \\
9,20,30,41 \\
8,20,30,42 \\
7,20,30,43
\end{cases}
\]

Active list = \{ P_{s1}^1, P_{s1}^1, P_{s1}^1, P_{s2}^1, P_{s2}^1, P_{s3}^2, P_{s3}^2, P_{s3}^2, P_{s3}^2 \}.

Proceeding in this way, we get

**Step- 7:**
This subproblem occurs at level 3 which is not equal to \{4\}. Hence, check the dominance property, \(\sigma = \{5,1,3\} \), \(\sigma' = \{2,4\} \), \(q_{s13} = q_{13} - t_5 = \)

\[
\begin{pmatrix}
-16,15, 43,74 \\
-29,15, 43,87 \\
-42,15, 43,100 \\
-55,15, 43,113
\end{pmatrix}
\]

\[
\begin{pmatrix}
4, 10,14, 20 \\
2,10,14, 22 \\
0,10,14, 24 \\
-2,10,14, 26
\end{pmatrix}
\]
\[q_\sigma = \begin{pmatrix} -36,1,33,70 \\ -51,1,33,85 \\ -66,1,33,100 \\ -81,1,33,115 \end{pmatrix}\]

\[\max_{i \in \sigma} d_i = \begin{pmatrix} -36,1,33,70 \\ -51,1,33,85 \\ -66,1,33,100 \\ -81,1,33,115 \end{pmatrix}\]. Since, the maximum value is equal to

\[q_\sigma = \begin{pmatrix} -36,1,33,70 \\ -51,1,33,85 \\ -66,1,33,100 \\ -81,1,33,115 \end{pmatrix}\].

**Step-8:**

Job 4 has an element in \(\sigma^1\) which has the highest due date. Hence based on the dominance property the subproblem \(P_{513}^4\) is further partitioned with a single branch \(P_{513}^4, \sigma = \{5,1,3\}\ & j = 4, q_\sigma = \begin{pmatrix} -36,1,33,70 \\ -51,1,33,85 \\ -66,1,33,100 \\ -81,1,33,115 \end{pmatrix}\), \(V_{j\sigma} = V_\sigma + \max (0, q_\sigma - d_j)\)

\[v = \begin{pmatrix} -110,5,113,228 \\ -148,5,113,266 \\ -186,5,113,304 \\ -224,5,113,342 \end{pmatrix} + \max \begin{pmatrix} -36,1,33,70 \\ -51,1,33,85 \\ -66,1,33,100 \\ -81,1,33,115 \end{pmatrix}\]

\[= \begin{pmatrix} -110,5,113,228 \\ -148,5,113,266 \\ -186,5,113,304 \\ -224,5,113,342 \end{pmatrix} \]

**TREE DIAGRAM**
The minimum total tardiness value is 59.
The required optimal sequence is 2→4→5→1→3.

CONCLUSION
We considered single machine scheduling problem (SMSP) with fuzzy processing time and fuzzy due date to minimize the total tardiness. This method is very easy to understand each stage that will help the decision maker in determining a best schedule for a given set of jobs effectively. This method has significant use of practical results in industries.

REFERENCES


