International Journal of Applied Science-Research and Review (IJAS) www.ijas.org.uk

Original Article

Bi-objective Parallel Machine Scheduling with Fuzzy Processing Time: Total Tardiness/Number of Tardy Jobs

Sameer Sharma[1], Seema Sharma[1], Deepak Gupta[2] and Naveen Gulati*[3]

[1]Department of Mathematics
D.A.V. College, Jalandhar, 144008, Punjab, India
[2]Department of Mathematics & Humanities
M. M. University, Mullana, Ambala, Haryana, India
[3]Department of Mathematics
S.D. College, Ambala Cantt., Haryana, India

ARTICLE INFO

Received 16 Jan. 2016
Received in revised form 04 Feb. 2016
Accepted 10 Feb. 2016

Keywords:
Fuzzy processing time;
Average high ranking;
Total tardiness;
Number of tardy jobs.

Corresponding author: Department of Mathematics S.D. College, Ambala Cantt., Haryana, India
E-mail address:
naveengulatimaths@gmail.com

ABSTRACT

In this paper we have considered the scheduling on parallel machines which optimizes the Total Tardiness and Number of Tardy jobs simultaneously. The processing times and setup times of jobs are uncertain in nature and only estimated values are given. The fuzzy triangular membership function is used to describe uncertainly involved. The objective of the paper is to find the optimal sequence of the jobs processing on parallel identical machines so as to minimize the secondary criteria of Total Tardiness with the condition that the primary criteria of number of tardy jobs remains optimized. The numerical illustrations are also given to demonstrate the computational efficiency of algorithm proposed.

INTRODUCTION

Scheduling is very common activity in both industry and non industry settings. Every day meetings are scheduled, maintenance and upgrade operations are planned, sports games are scheduled etc. If the jobs are properly scheduled then not only the efficiency of the jobs increases but the resources conflicts also prevented. The parallel machine scheduling problem is widely studied optimization problem. A schedule that optimizes one criterion may in fact perform quite poorly for another. Decision makers must carefully evaluate the trade-offs involved in considering several criteria. Bicriteria scheduling problems a subset of multi criterion scheduling problem are motivated by the fact that they are more meaningful from practical point of view. Bicriteria scheduling of jobs on identical parallel machines means jobs belong to two disjoint sets, and each set has a criterion to be optimized. The jobs are all available at time zero and have to be scheduled on m parallel machines.

Scheduling is an enduring process where the existence of real time information frequently forces the review and modification of pre-established schedules. The real world is complex and complexity generally arises from uncertainty. From this prospective the concept of fuzziness is introduced in the field of scheduling. Fuzzy sets and logic can be used to tackle uncertainty inherent in actual scheduling problem. In the present paper we have studied the bicriteria scheduling on parallel machines with fuzzy processing time involving total tardiness and number of tardy jobs. The problem is divided in two steps; in the primary step Number of Tardy jobs are calculated and in secondary step, Total Tardiness is minimized with the bi-objective function Total Tardiness/ Number of Tardy jobs.

A survey of literature has revealed a little work reported on the bicriteria scheduling problems on the parallel machines. Anghinolfi and Paolucci[1] studied total tardiness on parallel machines.. Parkash[7] studied the bi-criteria scheduling problems on parallel machines. Shim and Kim[10] dealt with scheduling on parallel identical machines to minimize the total tardiness. Sarin and Hariharan[5] considered the bicriteria problem of scheduling 'n' jobs on two machines to minimize the primary criterion of maximum tardiness and secondary criterion of number of tardy jobs. Sarin and Parkash [9] consider the problem of scheduling jobs on parallel identical machines so as to minimize primary and secondary criteria. Gupta *et al*[4] studied the scheduling on parallel machines with bi-objective function NT/T_{max} in fuzzy environment. Sunita and Singh[11] studied the bi-objective in fuzzy scheduling on parallel machines .Cenna and Tabucanon[2] studied bicriteria Scheduling problem in a job shop with parallel processor. Hariharan[5] developed bicriteria optimization of schedules on one and two machines. Martin and Roberto [6] studied Fuzzy scheduling with application to real time system. Sharma *et al*[12] studied the bicriteria scheduling on parallel machine involving total tardiness and weighted flow time in fuzzy environment. Sharma *et al*[13] studied the bi-objective problem with total tardiness and number of tardy jobs as primary and secondary criteria respectively for any number of parallel machines in fuzzy environment. Some of the noteworthy approaches are due to Zadeh[16], Gupta[3], Yager[14], Yao and Lin[15].

In general, the two approaches can be involved for bicriteria scheduling problems: both criteria are optimized simultaneously by using suitable weights for the criteria; secondly, the criteria are optimized sequentially by first optimizing the primary criterion and then the secondary criterion subject to the value obtained for the primary criterion. In the present paper, we study the bi-objective scheduling on parallel machines by executing second technique.

ROLE OF FUZZY LOGIC IN SCHEDULING

Scheduling is very important in real-time systems as it accomplishes the crucial goal of achieving a feasible schedule of the tasks. However, the uncertainty associated with the timing constrains of the real-time tasks makes the scheduling problem difficult to formulate. A fuzzy system can be thought of an attempt to understand a system for which no model exists, and it does so with the information that can be uncertain in a sense of being vague, or fuzzy, or imprecise, or altogether lacking. From this angle, fuzzy logic is a method to formalize the human capacity of imprecise reasoning. Such reasoning represents the human ability to reason approximately and judge under uncertainty. The real world is complex; complexity in the world generally arises from uncertainty. Zadeh [16] stated that most of the early interest in fuzzy set theory pertained to representing uncertainty in human cognitive system. Uncertainty can be thought of in an epistemological sense as being the inverse of information. Information about a particular problem may be incomplete, imprecise, fragmentary, unreliable, vague or deficient in some other way. From this prospective, the concept of fuzzy environment is introduced in the theory of scheduling.

BASIC FUZZY SET THEORY

A fuzzy set $\tilde{A}$ defined on the universal set of real numbers R, is said to be a fuzzy number if its membership function has the following characteristics:

(i) $\mu_{\tilde{A}} : R \to [0,1]$ is continuous.

(ii) $\mu_{\tilde{A}} = 0$ for all $x \in (-\infty, a_1) \cup (a_3, \infty)$

(iii) $\mu_{\tilde{A}}$ is strictly increasing on $[a_1, a_2]$ and strictly decreasing on $[a_2, a_3]$.

(iv) $\mu_{\tilde{A}} = 1$ for x = a_2.

Triangular Fuzzy Number

It is a fuzzy number represented with three points as follows: $\tilde{A} = (a_1, a_2, a_3)$, where a_1 and a_3 denote the lower and upper limits of support of a fuzzy set $\tilde{A}$. The membership value of the x denoted by $\mu(x), x \in R^+$, can be calculated according to the following formula.

$$\mu(x) = \begin{cases} 0; x \le a_1 \\ \dfrac{x - a_1}{a_2 - a_1}; a_1 < x < a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}; a_2 < x < a_3 \\ o; x \ge a_3 \end{cases}$$

Average High Ranking <A.H.R.>

The system characteristics are described by membership function; it preserves the fuzziness of input information. However, the designer would prefer one crisp value for one of the system characteristics rather than fuzzy set. In order to overcome this problem we defuzzify the fuzzy values of system characteristic, .i.e. Average High Ranking by using the Yager's [17] approximation formula

$$\text{Average High Ranking of } A(a_1,a_2,a_3) = h(A) = \frac{3a_2 + a_3 - a_1}{3}$$

Fuzzy Arithmetic Operations

If $A_1 = (m_{A_1}, \alpha_{A_1}, \beta_{A_1})$ and $A_2 = (m_{A_2}, \alpha_{A_2}, \beta_{A_2})$ be the two triangular fuzzy numbers, then

(i) $A_1 + A_2 = (m_{A_1}, \alpha_{A_1}, \beta_{A_1}) + (m_{A_2}, \alpha_{A_2}, \beta_{A_2}) = (m_{A_1} + m_{A_2}, \alpha_{A_1} + \alpha_{A_2}, \beta_{A_1} + \beta_{A_1})$

(ii) $A_1 - A_2 = (m_{A_1}, \alpha_{A_1}, \beta_{A_1}) - (m_{A_2}, \alpha_{A_2}, \beta_{A_2}) = (m_{A_1} - m_{A_2}, \alpha_{A_1} - \alpha_{A_2}, \beta_{A_1} - \beta_{A_2})$ if the following condition is satisfied $DP\left(\tilde{A}_1\right) \ge DP\left(\tilde{A}_2\right)$, where $DP(\tilde{A}_1) = \dfrac{\beta_{A_1} - m_{A_1}}{2}$ and $DP(\tilde{A}_2) = \dfrac{\beta_{A_2} - m_{A_2}}{2}$.

Here, DP denotes difference point of a Triangular fuzzy number.

Otherwise;

$A_1 - A_2 = (m_{A_1}, \alpha_{A_1}, \beta_{A_1}) - (m_{A_2}, \alpha_{A_2}, \beta_{A_2}) = (m_{A_1} - \beta_{A_2}, \alpha_{A_1} - \alpha_{A_2}, \beta_{A_1} - m_{A_2})$

(iii) $kA_1 = k(m_{A_1}, \alpha_{A_1}, \beta_{A_1}) = (km_{A_1}, k\alpha_{A_1}, k\beta_{A_1})$; if $k > 0$.

(iv) $kA_1 = k(m_{A_1}, \alpha_{A_1}, \beta_{A_1}) = (k\beta_{A_1}, k\alpha_{A_1}, km_{A_1})$; if $k < 0$.

The following notations will be used all the way through the present paper.

- i : ith job, $i = 1, 2, 3, \ldots, n$
- di : due date of the ith job
- ci : completion time of ith job
- wi : weight of ith job
- Ti : tardiness of the ith job = max $(ci - di, 0)$
- $Tmax$: maximum tardiness
- NT : number of tardy jobs
- J : location of ith job on machine k, where $j = 1, 2, 3, \ldots, n$
- n : total number of jobs to be scheduled
- m : total number of machines
- k : machine on which ith job is assigned at the jth position.

Theorem: A sequence S of jobs following early due date (EDD) order is an optimal sequence with minimum Total Tardiness.

.i.e. when jobs are processed on any of available parallel machines by early due date rule, the corresponding sequence of job processing is optimal with respect to minimum total tardiness as given by Sharma *et al* [12,13].

Algorithm

The following algorithm proposed to find the optimal sequence for bi-criteria problem Total Tardiness /NT.

Step 1: *Arrange the jobs in the increasing order of their due dates; break the ties arbitrarily.*

Step 2: *Calculate the number of tardy jobs i.e. NT.*

Step 3: *If $NT \leq m$ and all the late jobs are located at the identical locations on all the machines then the schedule is optimal; else, go to step 4.*

Step 4: *Find the first late job and put it in the late set L. If no more late job exists, then stop; else go to step 5.*

Step 5: *Allocate all the jobs j, $j \notin L$, using EDD rule as in step 1 and go to step 4.*

Step 6: *Arrange all the jobs of the late set L in EDD order and reassign them one by one to the earliest available machine.*

Step 7: *Designate all the jobs not covered in the set L of late jobs in the set E of jobs. The jobs in E are early. Arrange these jobs in EDD order.*

Step 8: *Pick the first job i from the set L. If none exist, then stop; else, find the earliest job j* such that $C_k+1 \leq d_k \ \forall \ k \in F_j$ where F_J *be the set of the followers of a job j on all the machines, with job j being inclusive in set* F_J and satisfied the given condition too. If none exist, then go to step10; else go to step9.

Step 9: Interchange the job *j* and *i* and set L = L – {*i*}; go to step8. If L is empty, then stop.

Step 10: Assign the jobs in set L to the machine after the jobs which have already been allocated, and exit.

Theorems

The following theorems have been developed to discuss the optimality of the proposed algorithm.

The proposed algorithm optimizes the total number of tardy jobs.

Proof: The proof is divided into two cases

Case 1: If $NT \leq m$, .i.e. number of tardy jobs is less than number of machines and all the late jobs are located at the identical locations on all the machines.

In order to prove the required result we firstly assume that there exists a schedule S_1 which is better than the schedule *S* obtained in the above algorithm. In S_1 all the late jobs of schedule *S* occupies an earlier position and the job having earlier due date moved to the next position. As the jobs are in EDD order some job must also be late. Therefore, by moving a late job to an earlier position more jobs may become late. So, sequence S_1 of jobs is not better than sequence *S* of jobs processing. This implies the *S* is optimal. Clearly the case of $NT=1$ is covered in this part itself.

Case 2: If $NT \geq 2$ and late jobs do not appear on the same location of the machine.

Following the first 5 steps of above algorithm, Consider the first late job *i* and put it in the late set. Let this move is not optimal and some other move can be made to obtain a better schedule. There is a possibility of two other moves: first, move job *i* to an earlier position and second, job *k* moves to the later position. We discuss these in two sub cases as follows:

Subcase1: When job *i* moves to an earlier position.

Now, if the job *i* is currently not late then after move, the new schedule has no additional late job or one or more jobs become late but if the late job is moved to the earlier position then as the jobs are in EDD order so, at least one of job remain late in new schedule. This implies the new schedule is not better than the schedule obtained by above algorithm.

Subcase2: When job *k* other than job *i* move to the later position.

Now, if job *k* is not currently late and also it remains early in new schedule then there will be no change in NT as jobs are in EDD order. But, if job is delayed in new schedule after move then the same improvement can also be made by moving the first late job to the late set as processing times are same.

Next, we consider the situation in which the late job, other than the first late job, is moved to the late set. Let this job be $j \neq i$. Let D_i be the set of late jobs after job *i* as obtained in the above algorithm. So, accordingly D_j be the set of late jobs after job *j*. Note that $D_j \subset D_i$.

Removing job j from the schedule and moving it to the late set affects the tardiness value of the jobs in D_j as the remaining jobs are rearranged in EDD order according to step 6. On the other hand, a similar movement of job i to the late set affects the tardiness of the jobs in D_i. Thus, any improvement that can be obtained by delaying job j can also be obtained by delaying job i. This implies the new schedule is not better than the schedule obtained by above algorithm.

The proposed algorithm optimizes the bi-criteria problem Total Tardiness/NT.

Proof: From the first five steps of above algorithm it is clear that jobs in the late set are the jobs having largest possible due dates. Now, we shall prove the efficiency of the proposed algorithm by testing the optimality of step 8 of the proposed algorithm.

In order to prove it, First we assume that the schedule given by the above algorithm say S is not optimal and there exit a schedule say S_1 which is more optimal. It implies somewhere in schedule S_1 , a job $i \in L$ is inserted before job j such that for some job $k \neq i \in F_j$ and $C_k+1 > d_k$. Now two cases arise in this condition.

Case 1: If job i is late in sequence S_1 of jobs, the number of tardy jobs increases which violates the primary criterion .Hence sequence S of jobs processing is optimal.

Case 2: If job i is early in sequence $S_1$of jobs, then there exit some job m such that $d_m \leq d_i$. Therefore, an improvement in Total Tardiness is possible by interchanging jobs m and i. Hence, Sequence S of jobs processing is at least as good as sequence S_1.

NUMERICAL ILLUSTRATIONS

Consider jobs with fuzzy processing time, due date on three parallel machines in a flow shop as given in table 1. The objective is to obtain a sequence of the jobs optimizing the bicriteria taken as Total Tardiness /NT.

Therefore, Total Tardiness = 26/3 units NT = {2}

Since NT $\leq m$ and all the late jobs are located at the identical locations on all the machines. Therefore, an optimal value for NT is achieved and minimum possible number of tardy jobs, NT=2.

Let L be the set of late jobs obtained above i.e. L = {3, 2} and E be the set of early jobs arranged in EDD i.e. E = {4, 1, 5}

Consider first Late job i = 3∈L and there is an earliest job j =5 in the schedule satisfying $C_j+1 \leq d_j$ where F_j = {5,3,2} such that $C_k+1 \leq d_k$ $\forall\ k \neq i \in F_j$. So interchange jobs i and j in the schedule. (table no. 3)

Therefore, Total Tardiness = 29/3 units and NT = {2}. Set $L = L$- {i} = {2}

Consider first Late job i = 2∈L, there is an earliest job j =3 in the schedule satisfying $C_j+1 \leq d_j$ where F_j = {3, 5, 2} such that $C_k+1 \leq d_k$ $\forall\ k \neq i \in F_j$. So interchange jobs i and j in the schedule.(table no. 4)

Therefore, Total Tardiness = 17/3 units and NT = {2}

Set L = L- {i} = {∅}

Therefore, the optimal level is achieved. Hence the optimal sequence of jobs processing is 4 – 1 – 2 – 3 – 5 with minimum total tardiness as 17/3 units and number of tardy jobs, NT= 2.(table no. 5)

Consider another example of jobs with fuzzy processing time, due date on three parallel machines in a flow shop as given in Table 6. The objective is to obtain sequence of the jobs optimizing the bicriteria taken as Total Tardiness /NT.

On arranging the jobs in EDD order on the parallel available machines M_1, M_2 and M_3,

We have (table no, 7)

Therefore, Total Tardiness = 71/3 units and NT = {4}

Now NT$\nleq m$ and also all the late jobs are not located at the identical locations on all the machines. Therefore NT is not optimal. For the optimality of NT, we proceed as follows:

So consider first late job 1.

Put it in the late set L = {1}.

Allocate all the jobs j; $j \notin L$. (table no. 8)

Now, consider first late job 2. Put it in the late set L = {1, 2}.

Allocate all the jobs j; $j \notin L$. (table no. 9)

Now consider first late job 6. Put it in the late set L = {1, 2, 6}.

Allocate all the jobs j; $j \notin L$ (table no. 10)

Now there exists no late job. Therefore, we have L = {1, 2, 6} and E be the set of early jobs arranged in EDD i.e. E = {5, 3, 4}. Now arrange all the jobs of late set in EDD, we have(table 11)

Now, NT = {3}

Since NT ≤ m and all the late jobs are located at the identical locations on all the machines. So, optimal value for NT is obtained.

Here, L be the set of late jobs obtained above i.e. L = {1, 3, 6} and E be the set of early jobs arranged in EDD i.e. E = {5, 4, 2}

Consider first job $i = 1 \in L$. Now ∃ earliest job $j = 6$ in the schedule satisfying $C_j+1 \leq d_j$ where $F_j = \{6\}$ such that $C_k+1 \leq d_k \;\; \forall\, k \neq i \in F_j$. Therefore, on interchange job i =1and job j=6 in the schedule, we have (table 12)

Therefore, Total Tardiness = 69/3 units, NT = {3} and $L = L- \{i\} = \{2, 6\}$

On considering first job $i = 2 \in L$, there is an earliest job $j = 6$ in the schedule satisfying $C_j+1 \leq d_j$ where $F_j = \{6\}$ such that $C_k+1 \leq d_k \;\; \forall\, k \neq i \in F_j$. So interchange the jobs i=2 and j=6 in the schedule. (table no. 13)

Therefore, Total Tardiness = 69/3 units, NT = {3} and $L = L- \{i\} = \{6\}$

Next, on considering first job $i = 6 \in L$, there exist no earliest job j such that $C_k+1 \leq d_k \;\; \forall\, k \in F_j$ where F_J be the set of the followers of a job j on all the machines, with job j being inclusive in set F_J and satisfied the given condition too. So assign the job 6 to the machine after the jobs which have already been allocated i.e. in the last(table no. 14)

Therefore, Total Tardiness = 69/3 units NT = {3}.

Therefore, the optimal level is achieved. Hence the optimal sequence of jobs processing is 5 – 3 – 4 – 2 – 1 - 6 with minimum total tardiness as 69/3 units and NT = 3.(table no.15)

CONCLUSION

In this paper, we have introduced the concept of fuzzy processing time in bicriteria scheduling on parallel machines to minimize the secondary criterion of total tardiness without violating the primary criterion of number of tardy jobs. The validity of the proposed method is explained with numerical illustrations. The proposed algorithm can be extended to n-jobs m-machines flow shop scheduling problem with uncertain parameters.

REFERENCES

1. Anghinolfi, D. and Paolucci, M. (2007), "Parallel machine total tardiness scheduling problem", *Computers and Operation Research*, 34(11) (2007) 3471-3490.
2. Cenna ,A.A. and Tabucanon ,M.T., " Bicriteria Scheduling Problem with minimum weighted Completion time and maximum allowable Tardiness", *International Journal of Production Economics*, 25(1 – 3) (1991) 95–101.
3. Gupta, J.N.D, Ruiztorres, A.J., Webser, S., "Minimization maximum tardiness and number of tardy jobs on parallel machines subject to Minimal flow time", *Journal of Operation Research Society*, 54 (2003) 1263 – 1274.
4. Gupta,D., Sharma,S., Aggarwal,S., "Bi-objective parallel machine scheduling with uncertain processing time", *Advances in Applied Science Research*, 3 (2012) 1020-1226.
5. Hariharan, R., "Bicriteria optimization of schedules on one and two machines", *Master Thesis*, Virginia Polytechnic Institute & State University (1988).
6. Martin, L. and Roberto, T., "Fuzzy scheduling with application to real time system", *Fuzzy sets and Systems*, 121(3) (2000) 523-535.
7. Parkash, D., "Bi-criteria Scheduling problems on parallel machines" *Ph.D. Thesis*, University of Birekshurg (1997) Virginia.
8. Sarin, S.C. and Hariharan, R., "A two machine bicriteria scheduling problem", *International Journal of Production Economics*, 65(2) (2000) 125-139.
9. Sarin, S.C. and Parkash, D., "Equal processing time bicriteria scheduling on parallel machines", *Journal of Combinatorial Optimization*, 8(2004), 227-240.

10. Shin, S.O., Kim, Y.D., "Scheduling on parallel machines to minimize total tardiness", *Journal of Operational Research*, 177(2007) 629-634.
11. Singh, T. P, and Sunita, "Bi-objective in fuzzy scheduling on parallel machines", *Arya Bhatta Journal of Mathematic and informatics*, 2 (2011) 149-152.
12. Sharma, S., Gupta, D. and Seema, "Bicriteria scheduling on parallel machines: total tardiness and weighted flowtime in fuzzy environment", *International Journal of Mathematics in Operational Research*, 5(4) (2013) 492-507.
13. Sharma, S., Gupta, D. and Seema, "Bi-Objective scheduling on parallel machines in fuzzy environment", *Advances in Intelligent System and Computing*, 236 (2013) 365-372.
14. Yager R.R., "A Procedure for ordering fuzzy subsets of unit interval", *Information Sciences*, 24(1981) 143 – 161.
15. Yao, J. S., Lin, F. T., "Constructing a fuzzy flowshop sequencing model based on statistical data", *International Journal of Appropriate Reasoning*, 29(3) (2002) 215-234.
16. Zadeh, L.A., "Fuzzy Sets", *Information and Control*, 8(1965) 78 - 98.

Table 1: Jobs with fuzzy processing time

Jobs	1	2	3	4	5
Processing time (in fuzzy env.)	(8,9,10)	(15,16,17)	(8,9,10)	(5,6,7)	(7,8,9)
Due time (d_i)	10	18	15	7	14

Solution: The AHR (Average High Ranking) of processing time of given jobs is as follow:

Table 2: Jobs with AHR of processing time

Jobs	1	2	3	4	5
AHR of processing time	29/3	50/3	29/3	20/3	26/3
Due time (d_i)	10	18	15	7	14

Table 3: Jobs scheduling following EDD order

Jobs	4	1	5	3	2
M_1	0 – 20/3			20/3 – 49/3	
M_2	0 – 29/3	M_3	0 – 26/3	26/3 – 76/3	
d_i	7	10	14	15	18
T_i	-	-	-	4/3	22/3

Table 4: Jobs flow table after interchanging jobs

Jobs	4	1	3	5	2
M_1	0 – 20/3			20/3 – 46/3	
M_2		0 – 29/3			29/3 – 79/3
M_3			0 – 29/3		
d_i	7	10	15	14	18
T_i	-	-	-	4/3	25/3

Table 5: Jobs flow table after interchanging job i and job j

Jobs	4	1	2	5	3
M_1	0 – 20/3			20/3 – 46/3	
M_2		0 – 29/3			29/3 – 58/3
M_3			0 – 50/3		
d_i	7	10	18	14	15
T_i	-	-	-	4/3	13/3

Table 6: Jobs with fuzzy processing time

Jobs	1	2	3	4	5	6
Processing time (in fuzzy env.)	(8,9,10)	(7,8,9)	(6,7,8)	(7,8,9)	(5,6,7)	(6,7,8)
Due time (d_i)	20/3	28/3	25/3	27/3	21/3	30/3

Solution: The AHR of processing time of given jobs is as follow:

Table 7: Jobs with AHR of processing time

Jobs	1	2	3	4	5	6
AHR of processing time	29/3	26/3	23/3	26/3	20/3	23/3
Due time (d_i)	20/3	28/3	25/3	27/3	21/3	30/3

Table 8: Jobs scheduling following EDD order

Jobs	1	5	3	4	2	6
M_1	0 – 29/3					29/3 – 52/3
M_2		0 – 20/3		20/3 – 46/3		
M_3			0 – 23/3		23/3 – 49/3	
d_i	20/3	21/3	25/3	27/3	28/3	30/3
T_i	9/3	-	-	19/3	21/3	22/3

Table 9: Jobs flow table

Jobs	5	3	4	2	6
M_1	0 – 20/3			20/3 – 46/3	
M_2		0 – 23/3			23/3 – 46/3
M_3			0 – 26/3		
d_i	21/3	25/3	27/3	28/3	30/3
T_i	-	-	-	18/3	16/3

Table10: Jobs flow Table

Jobs	5	3	4	6
M_1	0 – 20/3			20/3 – 43/3
M_2		0 – 23/3		
M_3			0 – 26/3	
d_i	21/3	25/3	27/3	30/3
T_i	-	-	-	13/3

Table 11: Jobs flow table

Jobs	5	3	4
M_1	0 – 20/3		
M_2		0 – 23/3	
M_3			0 – 26/3
d_i	21/3	25/3	27/3
T_i	-	-	-

Table 12: Jobs in EDD order on the parallel available machines

Jobs	5	3	4	1	2	6
M_1	0 – 20/3			20/3 – 49/3		
M_2		0 – 23/3			23/3 – 49/3	
M_3			0 – 26/3			26/3 – 49/3
d_i	21/3	25/3	27/3	20/3	28/3	30/3
T_i	-	-	-	29/3	21/3	19/3

Table 13: Jobs flow table after interchanging

Jobs	5	3	4	6	2	1
M_1	0 – 20/3			20/3 – 43/3		
M_2		0 – 23/3			23/3 – 49/3	
M_3			0 – 26/3			26/3 – 55/3
d_i	21/3	25/3	27/3	30/3	28/3	20/3
T_i	-	-	-	13/3	21/3	35/3

Table 14: Jobs flow table after interchanging

Jobs	5	3	4	2	6	1
M_1	0 – 20/3			20/3 – 46/3		
M_2		0 – 23/3			23/3 – 46/3	
M_3			0 – 26/3			26/3 – 55/3
d_i	21/3	25/3	28/3	28/3	30/3	20/3
T_i	-	-	-	18/3	16/3	35/3

Table 15: Jobs flow table after assigning job i to the last position

Jobs	5	3	4	2	1	6
M_1	0 – 20/3			20/3 – 46/3		
M_2		0 – 23/3			23/3 – 52/3	
M_3			0 – 26/3			26/3 – 49/3
d_i	21/3	25/3	28/3	28/3	20/3	30/3
T_i	-	-	-	18/3	32/3	19/3

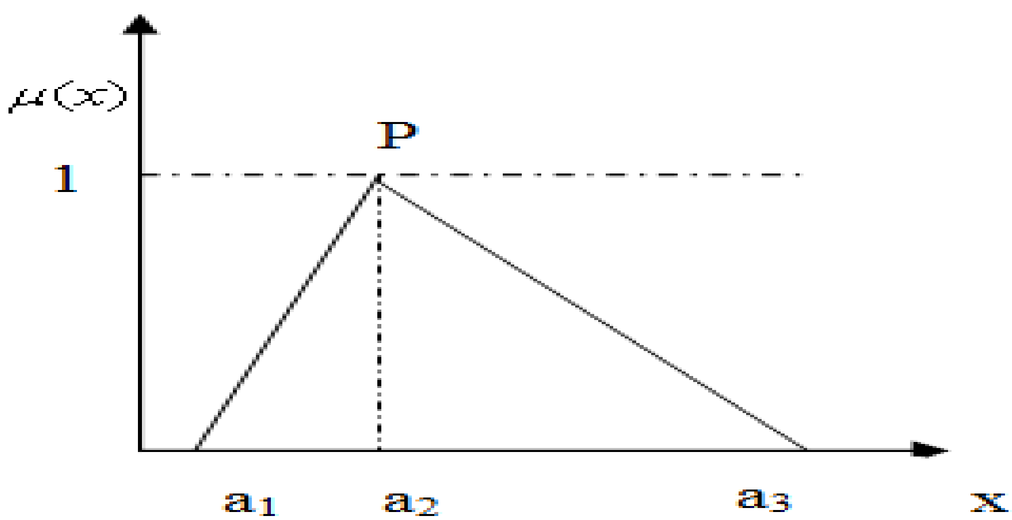

Fig. 1: Triangular fuzzy number